



**Fábio Luís Malhão
Verde**

**Sistema para Aquisição de Sinais Vitais com
Ligação a Smartphone**



**Fábio Luís Malhão
Verde**

**Sistema para Aquisição de Sinais Vitais com
Ligação a Smartphone**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações (Mestrado Integrado), realizada sob a orientação científica do Prof. Arnaldo Silva Rodrigues de Oliveira, Professor do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

o júri / the jury

presidente / president

Prof. Doutor Amaro Fernandes de Sousa

Professor Auxiliar da Universidade de Aveiro (por delegação da Reitora da Universidade de Aveiro)

vogais / examiners committee

Prof. Doutor Arnaldo Silva Rodrigues de Oliveira

Professor Auxiliar da Universidade de Aveiro (orientador)

Prof. Doutor João Agostinho Batista de Lacerda Pavão

Professor Auxiliar da Universidade de Trás-Os-Montes e Alto Douro (arguente)

agradecimentos / acknowledgements

É difícil imaginar a escrita desta dissertação sem a ajuda que recebi ao longo de todo este caminho. Gostaria de agradecer a todos os que me ajudaram das mais variadas formas.

Em primeiro lugar, um agradecimento muito especial à minha família e à Virginie Meira, pelo apoio, paciência e encorajamento prestados a todos os níveis.

Gostaria de agradecer ao Prof. Doutor Arnaldo Oliveira pela oportunidade de desenvolver este projeto, assim como toda a orientação e conhecimentos que me transmitiu.

À Prof. Doutora Odete Cruz e Silva e à Dra. Ilka Martins, pelas sugestões e observações relativas à aplicabilidade e usabilidade deste projeto.

Um agradecimento especial ao João Borges dos Santos pela boa atitude, disponibilidade e por toda a ajuda que me deu nos primeiros passos da dissertação.

Ao meu colega José Ferrão, pela disponibilidade e ajuda no desenvolvimento da aplicação Android.

Por último, mas não menos importante, gostaria de agradecer aos meus colegas Gabriel Sá Pinto, Walter Pires, Daniel Canedo e Henrique Gapo, pelo apoio e companhia nos momentos mais difíceis.

Resumo

Nas últimas décadas, a oximetria de pulso tem sido um dos mais significantes progressos tecnológicos na monitorização médica, com potencial para diversas aplicações clínicas. É uma técnica não-invasiva, simples e económica, fornecendo informações relativas ao ritmo cardíaco e ao nível de saturação de oxigénio no sangue (S_pO_2).

Pretende-se no âmbito desta dissertação o desenvolvimento de um sistema portátil de oximetria e temperatura que permita a extração, armazenamento, comunicação e análise de diversos sinais vitais, tais como a frequência cardíaca, o nível de saturação de oxigénio (S_pO_2) e a temperatura corporal de um utilizador.

Para desenvolver este projeto foi selecionado o MAX30102 como oxímetro do projeto. O circuito integrado deste sensor inclui os blocos de acondicionamento de sinal necessários para recolher toda a informação, tornando o circuito final mais simples. Este módulo possui ainda vários registos que permitem ao técnico, configurar o sensor para funcionar de diversas formas em termos de consumo energético, frequência de amostragem e resolução. Para se extrair a temperatura corporal do utilizador, foi usado o sensor de infravermelhos MLX90614. O CC1110, usado na fase inicial, e o Arduino Leonardo foram os dois microcontroladores usados no desenvolvimento deste projeto, com funções de recolha de amostras relativas ao ritmo cardíaco, S_pO_2 e temperatura corporal, assim como a construção e envio dos pacotes de dados para um módulo BLE (Bluetooth low energy). Ao microcontrolador do sistema (CC1110/Arduino), foi agregado o módulo BLE Bluefruit LE UART possibilitando a transferência dos pacotes de dados para um *smartphone Android*. Numa fase inicial, validou-se o sistema de aquisição dos sinais vitais com ligação ao MatLab no PC. Posteriormente, desenvolveu-se uma aplicação móvel *Android*, com o objetivo de receber e processar toda a informação referente aos sinais vitais e apresentar ao utilizador, em tempo real, o seu estado clínico.

O sistema de monitorização desenvolvido apresenta um elevado grau de correlação nos resultados de ritmo cardíaco e S_pO_2 relativamente a dois oxímetros comerciais utilizados (GIMA OXY-6 e CA-MI O2-easy) durante os testes. Em relação com o GIMA OXY-6, o sistema possui um erro de $\pm 2.65\%$ e de $\pm 0.28\%$ nos valores de ritmo cardíaco e S_pO_2 , respetivamente. Em comparação com o CA-MI O2-easy, o sistema possui um erro de $\pm 1.64\%$ e de $\pm 0.29\%$ nos valores de ritmo cardíaco e S_pO_2 , respetivamente.

A utilização do protótipo desenvolvido, fornece ao utilizador a possibilidade de monitorizar a sua atividade cardíaca e temperatura corporal de uma forma simples e fiável.

Abstract

In last decades, pulse oximetry has been one of the most significant technological advances in medical monitoring, with potential for diverse clinical applications. Characterized by being non-invasive, simple and economic, this technique provides real time oxygen saturation level (S_pO_2) and heart rate information.

It is intended in the context of this thesis the development of a modular oximetry system to allow the measurement, storing, reporting and analysis of various vital signs such as the heart rate, oxygen saturation level (S_pO_2) and the body temperature of a user.

To develop this project, the MAX30102 was chosen as the oximeter sensor. This sensor, has already all the signal blocks needed to collect all the data, making a cleaner circuit. This module has also several registers allowing the technical staff to configure the sensor to operate in different ways in terms of energy consumption, sampling rate and resolution. To extract the body temperature, the infrared sensor MLX90614 was used. The CC1110, used in the beginning of the project, and the Arduino Leonardo were used in this project and they had to receive samples, build and send data packages. A Bluetooth module was added to the microcontroller used (CC1110/Arduino) in order to enable communications to an Android mobile device. This system was first validated using the MatLab and after that, was developed an *Android* mobile application, in order to receive and treat all information concerning the vital signs, showing in real time the clinical state of an user.

The developed system presents a high degree of correlation in the results of heart rate and S_pO_2 compared to two commercial oximeters (GIMA OXY-6 e CA-MI O2-easy), used during the tests. Regarding the GIMA OXY-6, the system has an error of $\pm 2.65\%$ and $\pm 0.28\%$ in the values of heart rate and S_pO_2 , respectively. Regarding the CA-MI O2-easy, the system presents an error of $\pm 1.64\%$ and $\pm 0.29\%$ in the values of heart rate and S_pO_2 , respectively.

Thus, with this prototype, the user can easily monitor his heart activity, S_pO_2 and body temperature in a reliable way.

Conteúdo

Conteúdo	i
Lista de Figuras	v
Lista de Tabelas	ix
Lista de Acrónimos	xi
1 Introdução	1
1.1 Enquadramento	1
1.2 Motivação	1
1.3 Objectivos	2
1.4 Estrutura do documento	3
2 Conceitos fundamentais	5
2.1 Atividade cardíaca	5
2.1.1 Ritmo cardíaco	6
2.1.2 Doenças cardiovasculares	6
2.2 Hemoglobina e saturação de oxigénio	6
2.2.1 História da medição de SpO2	7
2.2.2 Método atual de medição de SpO2	8
2.2.3 Calibração	9
2.2.4 Fontes de erro	10
2.3 Oximetria de pulso	11
2.3.1 Princípios de operação	11
2.3.2 Tipos de sondas e modelos comerciais	12
Sonda de transmissão para o dedo	12
Sonda de transmissão para a orelha	13
Sonda de reflexão para a testa	13
Oxímetros no mercado	14
3 Dispositivos móveis	17
3.1 <i>Internet of things</i>	17
3.1.1 <i>Wearable devices</i>	18
3.1.2 Serviços de saúde	19
3.2 Smartphones	21
3.2.1 <i>Android</i>	21

3.2.2	<i>Bluetooth</i>	22
	<i>Bluetooth low energy</i>	23
4	Especificação e arquitetura do sistema	25
4.1	Introdução	25
4.2	Requisitos	25
4.3	Arquitetura do sistema	26
4.3.1	Oxímetro	27
4.3.2	Sensor de temperatura	27
4.3.3	Microcontrolador	27
4.3.4	Módulo BLE	28
5	Implementação do módulo de sensores	29
5.1	Sistema proposto	29
5.1.1	Oxímetro - MAX30102	30
5.1.2	Sensor de temperatura - MLX90614	30
5.1.3	Microcontroladores	30
	CC1110	30
	Arduino	31
5.1.4	Módulo BLE - Bluefruit LE UART	31
5.2	Interligação entre MCU, sensores e módulo BLE	32
5.2.1	Oxímetro	32
5.2.2	Sensor de temperatura	33
5.2.3	Módulo BLE	35
5.3	Funcionamento do oxímetro	36
5.3.1	Configurações	36
	FIFO	36
	Modo de funcionamento	36
	Modo SpO2	37
	LEDs	37
5.3.2	Aquisição das amostras do oxímetro	38
5.4	Construção e envio dos pacotes de dados	39
5.4.1	Envio de dados para o MatLab	39
5.4.2	Envio de dados para o módulo BLE	40
5.5	Sequência de operações do MCU	40
5.6	Estrutura de ficheiros	41
5.6.1	Ficheiros do CC1110	41
5.6.2	Ficheiros do Arduino	42
6	Desenvolvimento da aplicação móvel	43
6.1	Fundamentos da programação <i>Android</i>	43
6.1.1	Atividades	43
6.1.2	Ficheiro XML - <i>layout</i>	44
6.1.3	<i>Manifest</i>	45
6.1.4	Gradle	45
6.1.5	GAP	45
6.1.6	GATT	45

	Serviços	46
	Características	46
6.2	<i>Bluetooth low energy</i>	46
6.2.1	Configurações	47
6.2.2	Ligação com outros dispositivos	47
6.3	Receção dos dados	48
6.3.1	Aumento do MTU	48
6.3.2	Otimização dos dados	49
6.4	Visualização dos dados em tempo real	49
6.5	Determinação do ritmo cardíaco e SpO2	50
6.5.1	Metodologia	50
	Implementação dos filtros	52
	Calibração dos níveis de SpO2	54
6.6	Sequência de operações da aplicação <i>Android</i>	55
6.7	Estrutura de ficheiros	56
7	Resultados e discussão	57
7.1	Testes de configuração do oxímetro	57
7.1.1	Configuração do modo SpO2	57
7.1.2	Configuração dos LEDs	61
7.2	Funcionamento do sistema	62
7.3	Resultados experimentais	64
7.3.1	Medição no dedo	64
	Tratamento estatístico	65
7.3.2	Medição na testa	71
7.4	Análise de resultados	71
8	Conclusão e trabalho futuro	73
8.1	Conclusão	73
8.2	Trabalho futuro	73
	Bibliografia	75
A	<i>I₂C</i> - Bit banging	81
B	IAR Embedded Workbench	83
B.1	Criação e configuração de um projeto	83
B.2	Como compilar	84
C	Sistema com MatLab	87
C.1	Diagrama de blocos	87
C.2	Resultados	88
D	Criação da aplicação <i>Android</i>	89
D.1	Android Studio	89
D.1.1	Criação de um projeto	89
D.1.2	Interface com o utilizador	90
D.1.3	Como compilar e executar	91

Lista de Figuras

2.1	Sistema circulatório	5
2.2	Estrutura da hemoglobina	7
2.3	Curvas de absorção de espectro ótico por parte da Hb e HbO_2	8
2.4	Variações na atenuação da luz pelos vários tecidos	9
2.5	Relação empírica e modelo de Beer-Lambert entre o S_pO_2 e o rácio R	10
2.6	Oxímetro de pulso	11
2.7	Sonda de transmissão (esquerda) e sonda de reflexão (direita)	12
2.8	Sonda de transmissão para o dedo	13
2.9	Sonda de transmissão para a orelha	13
2.10	Sonda de transmissão para a testa	13
2.11	Oxímetro de pulso atual	15
3.1	<i>Internet of Things</i>	17
3.2	Sensor flexível de ECG com comunicação <i>wireless</i>	18
3.3	Dispositivo e aplicação <i>fitness</i>	18
3.4	Conjuntos de <i>wearable devices</i>	19
3.5	Sistema remoto de saúde genérico	19
3.6	Arquitetura do CardioSentinal	20
3.7	Interface gráfica para desenvolvimento das aplicações <i>Android</i>	22
3.8	Produção de dispositivos <i>Bluetooth</i> no planeta (mil milhões)	22
4.1	Diagrama de blocos do sistema	26
5.1	Diagrama do sistema proposto	29
5.2	Diagrama do MAX30102	30
5.3	Interfaces de comunicação	32
5.4	Configuração de um registo do MAX30102	33
5.5	Leitura de vários bytes de dados do MAX30102	33
5.6	Configuração de um registo do MLX90614	34
5.7	Leitura de dados do MLX90614	34
5.8	Formato da trama RS-232 usada	35
5.9	Registo FIFO	36
5.10	Registo modo de funcionamento	36
5.11	Registo S_pO_2	37
5.12	Registos do LED vermelho e infravermelho	37
5.13	Organização dos dados no FIFO do oxímetro	38
5.14	Envio dos pacotes de dados relativos à hemoglobina para o MatLab	39

5.15	Envio de pacotes de dados relativos à hemoglobina para o módulo BLE	40
5.16	Fluxograma das ações do MCU	41
6.1	Ciclo de vida de uma atividade	44
6.2	Especificações do BLE GATT	46
6.3	Estrutura dos pacotes de dados otimizados	49
6.4	Sinal PPG gerado pelas amostras do oxímetro	51
6.5	Etapas do processamento de sinal	51
6.6	Respostas em frequência do filtro passa-baixo	52
6.7	Sinal original e sinal após filtro passa-baixo	52
6.8	Respostas em frequência do filtro passa-alto	53
6.9	Sinal original e sinal após filtro passa-alto	53
6.10	Modelo de calibração quadrático	54
6.11	Curva de calibração linearizada	54
6.12	Fluxograma das ações da aplicação <i>Android</i> desenvolvida	55
7.1	Escala ADC 16384 nA e 15 bits de resolução	58
7.2	Escala ADC 16384 nA e 16 bits de resolução	58
7.3	Escala ADC 16384 nA e 17 bits de resolução	59
7.4	Escala ADC 16384 nA e 18 bits de resolução	59
7.5	Escala ADC 8192 nA e 15 bits de resolução	59
7.6	Escala ADC 8192 nA e 16 bits de resolução	59
7.7	Escala ADC 8192 nA e 17 bits de resolução	60
7.8	Escala ADC 8192 nA e 18 bits de resolução	60
7.9	Escala ADC 4096 nA e 18 bits de resolução	60
7.10	Escala ADC 2048 nA e 18 bits de resolução	60
7.11	Correntes dos LEDs - 6.4 mA	61
7.12	Correntes dos LEDs - 9.8 mA	61
7.13	Correntes dos LEDs - 12.5 mA	61
7.14	Correntes dos LEDs - 15.9 mA	61
7.15	Correntes dos LEDs - 19.3 mA	62
7.16	Alimentações da SmartRF04EB (esquerda) e do Arduino (direita)	62
7.17	Interruptor da SmartRF04EB (esquerda) e <i>reset</i> do Arduino (direita)	62
7.18	Tela de <i>scan</i> de dispositivos BLE no <i>smartphone</i>	63
7.19	Telas de exibição de resultados	63
7.20	Protótipo laboratorial com CC1110 (esquerda) e com Arduino (direita)	64
7.21	Sinal recolhido através do dedo	64
7.22	Extração de amostras com o sistema desenvolvido e o GIMA OXY-6	65
7.23	Gama das amostras de ritmo cardíaco dos três sistemas	67
7.24	Gráfico da dispersão dos resultados de ritmo cardíaco dos três sistemas	68
7.25	Gama das amostras de S_pO_2 dos três sistemas	69
7.26	Gráfico da dispersão dos resultados de S_pO_2 dos três sistemas	70
7.27	Sinal recolhido através da testa	71
A.1	Conexão I^2C comum	81
A.2	Sequências de <i>start</i> e <i>stop</i>	81
A.3	SDA e SCL durante a transferência de 1 byte de dados	82

A.4	SDA e SCL durante a transferência do endereço do <i>slave</i>	82
B.1	IAR Workbench - <i>Create project</i>	83
B.2	IAR Workbench - <i>options</i>	84
B.3	IAR Workbench - <i>debug</i>	84
C.1	Diagrama do sistema com ligação ao MatLab	87
C.2	Sinal cardíaco no MatLab	88
C.3	Apresentação dos resultados no MatLab	88
D.1	Configuração de um novo projeto	89
D.2	<i>Workspace</i> do Android Studio	90
D.3	Botão de <i>run</i>	91
D.4	Janela de escolha do dispositivo	91

Lista de Tabelas

2.1	Oxímetros disponíveis no mercado	14
5.1	Configurações da UART	35
7.1	Configurações da escala ADC	57
7.2	Configurações da resolução das amostras	58
7.3	Resultados das medições	66
7.4	Estatísticas sobre as amostras de ritmo cardíaco	67
7.5	Correlações de ritmo cardíaco entre os três sistemas	68
7.6	Estatísticas sobre as amostras de S_pO_2	69
7.7	Correlações de S_pO_2 entre os três sistemas	70

Lista de Acrónimos

AC	Alternating current
AAT	Attribute protocol
ADC	Analog-to-digital conversion
API	Application programming interface
AVD	Android virtual device
BLE	Bluetooth low energy
BPM	Batimentos por minuto
CTS	Clear to send
DC	Direct current
ECG	Eletrocardiograma
EEPROM	Electrically-erasable programmable read-only memory
FIFO	First-in first-out
FIR	Finite impulse response
GAP	Generic access profile
GATT	Generic attribute profile
GPIO	General-purpose input/output
GPS	Global positioning system
GSM	Global system for mobile communication
Hb	Hemoglobina
HbO2	Hemoglobina rica em oxigénio
HR	Ritmo cardíaco
I/O	Input/output
I2C	Inter-integrated circuit

IC	Circuito integrado
ID	Identification
IDE	Ambiente de desenvolvimento integrado
IoT	Internet of Things/Internet das Coisas
JDK	Java development kit
LAN	Local area network
LCD	Liquid crystal display
LED	Light emitting diode
LSb	Bit menos significativo
MAC	Media access control
MCU	Microcontroller unit
MSb	Bit mais significativo
MTU	Maximum transmission unit
O₂	Molécula de oxigénio
PC	Computador pessoal
PCB	Printed circuit board
PPG	Photoplethysmogram
RAM	Random-access memory
RTS	Read to send
RX	Receiver
SCL	Serial clock line
SDA	Serial data line
SDK	Software development kit
SNR	Signal-to-noise ratio
SPI	Serial peripheral interface
SPO₂	Saturação de oxigénio no sangue
TX	Transmitter
UART	Universal asynchronous receiver/transmitter
UI	Interface de utilizador

USB	Universal serial bus
UUID	Universally unique identifier
XML	Extensible markup language

Capítulo 1

Introdução

1.1 Enquadramento

A Europa é o continente do mundo com a população mais envelhecida. Este fenómeno, causado pelo aumento da esperança média de vida e pela diminuição do número da natalidade, traduz-se numa sociedade menos ativa. Tendo conhecimento das dificuldades que alguns idosos têm em viver independentemente, há a necessidade de manter uma constante vigilância das suas atividades e dos seus sinais vitais [ONU15].

Cada ano, 17.3 milhões de mortes são causadas por doenças cardiovasculares em todo o mundo, número que se prevê que continue a aumentar. Em 2013, as mortes por acidentes cardiovasculares representaram cerca de 31% das mortes no planeta, estimando-se que direta e indiretamente o gasto total neste tipo de doença é de 316.6 mil milhões de dólares [Ass16].

A constante recolha de dados vitais de qualquer utilizador, especialmente de idosos e pessoas com doenças cardiovasculares, tem-se tornado uma atividade cada vez mais usual. No passado, coletar este tipo de informação era dispendioso e difícil, porém, com o aumento do nível de integração de sensores, sistemas de processamento e de comunicação, é possível criar dispositivos de monitorização compactos e não-invasivos. Estes aparelhos permitem uma constante vigilância do estado clínico de um indivíduo nas mais variadas situações, aumentando assim os cuidados de saúde fornecidos à população [MZH15].

Devido às vantagens em termos de prevenção e controlo, os sistemas de monitorização de sinais vitais têm sido utilizados em áreas militares, desportivas e hospitalares. A possibilidade de visualizar remotamente sinais como o ritmo cardíaco e S_pO_2 em dispositivos móveis (e.g. *smartphone*), permite diminuir o tempo de resposta em situações de emergência, sendo este o contexto de trabalho desta dissertação.

1.2 Motivação

O ritmo cardíaco e S_pO_2 , medidos através da oximetria, são dois dos sinais biomédicos mais importantes na monitorização de pacientes; presentes em qualquer sala de cirurgia ou ambulância, auxiliam tanto no acompanhamento pré-operatório como no acompanhamento pós-operatório [Lem16].

O número de utilizadores de *smartphone* está a crescer constantemente, rondando os 2.1 mil milhões (cerca de 30% da população mundial), fazendo prever que até 2020 o número de utilizadores deste dispositivo aumente em cerca de mil milhões [Ols08]. Aproveitando este

mercado e o facto da população estar familiarizada com o uso deste equipamento, o sistema a desenvolver nesta dissertação enviará para o *smartphone*, em tempo real, toda a informação relativa aos sinais vitais mencionados. Desta forma, o utilizador tem ao seu dispor uma maneira simples e rápida de consultar o seu estado clínico.

É importante desenvolver um sistema capaz de recolher dados relativos ao ritmo cardíaco, S_pO_2 e temperatura corporal, assim como um sistema móvel e compacto de rápida, simples e fiável utilização para que qualquer utilizador possa monitorizar em qualquer lugar o seu estado clínico. Esta dissertação visa precisamente o desenvolvimento de um sistema com essas características.

1.3 Objectivos

Este projeto tem como objetivo principal o desenvolvimento de um sistema não-invasivo de aquisição do sinal cardíaco e temperatura corporal com conexão a um dispositivo móvel, para consulta do estado clínico de um indivíduo em tempo real. O protótipo a desenvolver deve apresentar as seguintes características:

- **Portátil** - permitindo ao utilizador monitorizar em qualquer lugar os seus sinais vitais;
- **Protocolos de comunicação abertos** - são protocolos padrão, acessíveis por qualquer técnico;
- **Extensível** - para possibilitar a integração de novos módulos de sensores, aumentando as funções do serviço de monitorização;
- **Baixo custo** - deve ser utilizado *hardware standard*, para criar um protótipo económico;
- **Tempo real** - para apresentação dos dados constantemente.

Relativamente à ordem de trabalhos, o projeto focar-se-á em:

- **Fase inicial** - passará por testar os módulos de oximetria e temperatura adquiridos e validar o funcionamento do sistema com conexão ao MatLab no PC (computador pessoal);
- **Fase intermédia** - será implementada uma comunicação BLE (*Bluetooth low energy*) e desenvolvida uma aplicação móvel para visualização, em tempo real, do estado clínico do utilizador;
- **Fase final** - deverão ser realizados testes a um grupo de indivíduos para comparar o desempenho do sistema desenvolvido relativamente a oxímetros comerciais.

1.4 Estrutura do documento

Além do capítulo introdutório, este documento está dividido em 7 capítulos com o seguinte conteúdo:

- No **capítulo 2** (“Conceitos fundamentais”) é feito um enquadramento teórico e técnico sobre o funcionamento do oxímetro de pulso, introduzindo a terminologia importante relativa à atividade cardíaca;
- O **capítulo 3** (“Dispositivos móveis”) sumaria o atual estado de arte da computação móvel contextualizando com cenários médicos. É ainda apresentada uma noção acerca da IoT (*Internet of Things*/Internet das Coisas), assim como dos *wearable devices* nos tempos atuais;
- O **capítulo 4** (“Especificação e arquitetura do sistema”) descreve de uma forma genérica a arquitetura do sistema desenvolvido, assim como as funções que se pretende que cada bloco desempenhe;
- O **capítulo 5** (“Implementação do módulo de sensores”) informa o leitor sobre cada módulo utilizado, como são agregados ao sistema final e como fluem os dados entre os vários blocos. Os processos de configuração, recolha e envio dos dados são abordados neste capítulo;
- O **capítulo 6** (“Desenvolvimento da aplicação móvel”) contém toda a informação acerca do desenvolvimento da aplicação *Android*, como foi realizado o tratamento das amostras recebidas e a extração dos valores relativos ao ritmo cardíaco e S_pO_2 ;
- O **capítulo 7** (“Resultados e discussão”) fornece informação acerca dos testes realizados ao módulo de oximetria, em termos dos seus parâmetros configuráveis. Além disso, são apresentados os testes efetuados a diferentes indivíduos com o sistema desenvolvido e com dois oxímetros de pulso comerciais. Por fim, é realizada uma discussão dos resultados obtidos;
- No **capítulo 8** (“Conclusão e trabalho futuro”) é concluída a análise dos resultados do capítulo anterior, avaliando o estado atual da dissertação. São apresentadas ainda algumas sugestões sobre futuros trabalhos, que visam o aperfeiçoamento do sistema desenvolvido.

Capítulo 2

Conceitos fundamentais

A monitorização de sinais vitais é essencial para qualquer utilizador de *wearable devices* determinar num dado instante a sua situação clínica. Esta função é tipicamente realizada por aparelhos capazes de simultaneamente exibir o ECG (eletrocardiograma), medir o ritmo cardíaco, S_pO_2 e temperatura corporal.

2.1 Atividade cardíaca

O oxigénio é um elemento essencial para a vida humana. Um fornecimento adequado de oxigénio aos órgãos vitais é de extrema importância, já que estes, necessitam deste componente para desempenhar as suas funções. Por outro lado um fraco fornecimento de oxigénio pode causar danos irreversíveis no cérebro e em último caso, a morte [Joh97].

Um adulto tem em média 5 litros de sangue a correr nos seus vasos sanguíneos e a sua função assenta no transporte de oxigénio, nutrientes e hormonas por todo o corpo. A escassez de sangue poderia causar o colapso de todo o sistema, por isso é que o coração desempenha um papel muito importante no corpo humano, gerindo o sistema circulatório e realizando as trocas de sangue com o sistema respiratório [Joh97, Cas07]. A figura 2.1 representa a ligação entre estes dois sistemas, fechando todo o ciclo cardíaco.

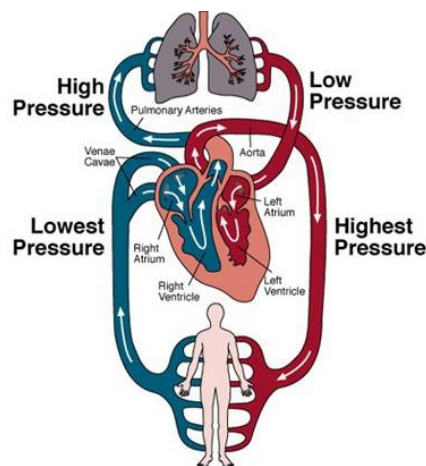


Figura 2.1: Sistema circulatório [Fer09].

2.1.1 Ritmo cardíaco

Assim como os restantes órgãos, o coração também precisa de uma certa quantidade de oxigénio para um funcionamento adequado.

Durante os primeiros anos de vida o organismo humano tem necessidade de uma maior quantidade de oxigénio, com o ritmo cardíaco a rondar os 120 BPM (batimentos por minuto). Aos 7 anos o ritmo cardíaco normal é de 85 BPM e finalmente aos 18 anos estabiliza nos 70 BPM [OPH⁺11]. No entanto, estes valores padrão de frequência cardíaca dependem de outros fatores, nomeadamente [Wat06]:

- Exercício físico;
- Drogas;
- Medicação;
- Esforço mental;
- Ruído.

2.1.2 Doenças cardiovasculares

O coração desempenha um papel fundamental no sistema humano mas quando a sua ação de bombear sangue não é regular ou é interrompida vários problemas podem surgir. O problema mais comum é o batimento irregular do coração, chamado de arritmia, que faz com que o coração bombeie sangue mais depressa ou mais devagar do que o normal. Esta doença é normalmente causada por instabilidades na química do sangue, uso de drogas ou por níveis hormonais irregulares [Nat16]. Outro tipo de problema muito comum é a doença arterial coronária, tão grave que pode originar um ataque cardíaco. Um ataque desta natureza acontece quando uma placa rica em colesterol explode e liberta o seu conteúdo para a corrente sanguínea, fazendo com que um coágulo de sangue se forme, bloqueando assim o fluxo sanguíneo [San05].

Por estas razões é importante manter constante atenção a estes fenómenos, realizando exames pelo menos uma vez por ano, ter uma alimentação saudável, não fumar nem consumir drogas, praticar desporto, controlar o colesterol, beber água regularmente e monitorizar os sinais vitais com regularidade [San05].

2.2 Hemoglobina e saturação de oxigénio

A hemoglobina (figura 2.2) é uma molécula que está presente no sangue e circula por todo o organismo através das artérias e vasos sanguíneos. Stokes deu um grande contributo para a ciência quando descobriu que esta molécula, existente nos glóbulos vermelhos, é uma proteína responsável pelo transporte de oxigénio ao longo do corpo [Klo03, Mai13].

O aumento da quantidade de glóbulos vermelhos no sangue é geralmente desencadeado pela prática de exercício físico ou adaptações a alta altitude uma vez que, o aumento deste tipo de glóbulos favorece o transporte de oxigénio por todo o sistema, melhorando o desempenho e resistência de um indivíduo [Klo03].

Cada molécula de hemoglobina é constituída por quatro cadeias de peptídeos e por quatro moléculas em forma de disco chamadas de *hemes* (figura 2.2). No centro de cada grupo de

hemes, encontra-se um átomo de ferro que pode ser combinado com uma molécula de oxigénio. Assim sendo, uma hemoglobina pode transportar quatro moléculas de oxigénio [Fox11].

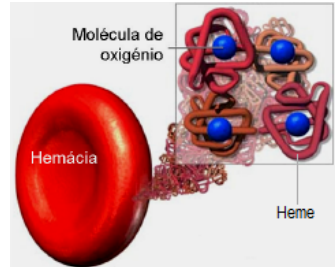


Figura 2.2: Estrutura da hemoglobina [RD09].

Existem duas formas de hemoglobina no sangue: hemoglobina pobre em oxigénio (*Hb*) existente em maior quantidade no sangue venoso e a hemoglobina oxigenada (*HbO₂*) existente no sangue arterial e formada nos pulmões pela junção *Hb* + *O₂* (molécula de oxigénio) [Sch08].

O *S_pO₂* exprime a quantidade de oxigénio que é transportada pelo sangue nas artérias e é representado pela razão entre a concentração de *HbO₂* e toda a hemoglobina presente no sangue, expressa pela equação 2.1 [Joh97].

$$SpO_2 = \frac{[HbO_2]}{[HbO_2] + [Hb]} \times 100(\%) \quad (2.1)$$

Por ser um dos sinais vitais mais importantes, é necessário avaliar regularmente os níveis de *S_pO₂* de um indivíduo. De uma forma geral, um ser humano saudável possui níveis de *S_pO₂* entre 95% e 100%, consequentemente, valores abaixo de 90% são considerados de risco, sendo aconselhável tratamento imediato [Wor11, SM08]. A monitorização deste sinal vital pode ser realizada através da oximetria de pulso (e.g); uma técnica não-invasiva, prática e possível de incorporar em *wearable devices*.

2.2.1 História da medição de SpO2

Os primeiros estudos na oximetria são atribuídos a Lambert por desenvolver, em 1760, uma relação entre a intensidade de luz absorvida e a quantidade de material absorvente; experiência posteriormente retomada por Beer em 1852 dando origem à lei de Beer-Lambert que representa a relação entre a intensidade de luz transmitida e incidente numa solução homogénea (equação 2.2) [TL15, Fox11], onde:

- *I_{trans}* - intensidade de luz transmitida;
- *I₀* - intensidade de luz incidente;
- $\varepsilon(\lambda)$ é - coeficiente de extinção de soluto;
- *C* - concentração de soluto;
- *D* - distância do percurso ótico.

$$I_{trans} = I_0 \cdot e^{-\epsilon(\lambda)CD} \quad (2.2)$$

Em 1864, usando um espectrómetro que acabara de ser inventado, Hoppe Seyler identificou a molécula de Hemoglobina e já em 1931, Nicolai Ludwyn iniciou uma investigação ligada à espectrofotometria quantitativa da luz transmitida pela pele humana [TL15].

Após a descoberta da hemoglobina, concluiu-se que a sua absorção da luz variava de acordo com a quantidade de oxigénio que transporta, uma vez que a HbO_2 é caracterizada por uma cor vermelha, já a Hb possui uma cor mais escura; a este princípio dá-se o nome de espectrofotometria. Este fenómeno acontece uma vez que a Hb e a HbO_2 têm uma absorção espectral de luz diferente nos comprimentos de onda entre 600 nm (luz vermelha) e 1000 nm (infravermelhos), como está representado na figura 2.3 [Joh97].

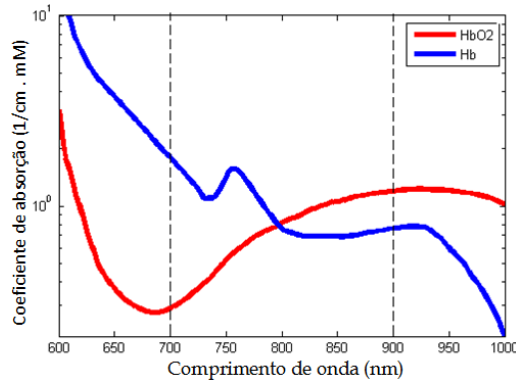


Figura 2.3: Curvas de absorção de espectro ótico por parte da Hb e HbO_2 [oH16].

Apesar da equação 2.2 ter sido usada originalmente para calibrar os oxímetros, não deve ser aplicada à oximetria, já que o sangue possui vários absorventes (solução não homogénea) [Joh97]. Para além do sangue, outras variáveis biológicas do corpo humano como a pele, músculos ou ossos também dispersam a luz, fazendo com que todos estes fatores não se possam correlacionar com a lei de Beer-Lambert.

2.2.2 Método atual de medição de SpO_2

A necessidade de se criar um método que fornecesse uma melhor aproximação do que a lei de Beer-Lambert incentivou novas pesquisas nesta área. Takuo Aoyagi foi o primeiro a aperceber-se que as variações da luz recolhidas através dos dedos, causadas pela atividade cardíaca, poderiam ser utilizadas para extrair os níveis de S_pO_2 . Em 1972, desenvolveu um oxímetro incorporando dois comprimentos de onda diferentes (vermelho e infravermelho) que media a componente oscilante do sangue nas artérias e retirava o valor do S_pO_2 .

Os oxímetros de pulso mais recentes dependem do facto da Hb e HbO_2 possuírem uma absorção espectral de luz diferente (como já referido) mas também de um método chamado de pletismografia ótica. Esta técnica de medição da quantidade de luz transmitida/refletida pelo sangue tem em consideração as flutuações no volume de sangue nas artérias, chamado de sinal PPG (*photoplethysmogram*), durante o ciclo cardíaco [Vij89].

Como mostrado na figura 2.4, a magnitude do sinal PPG depende da quantidade de sangue bombeado pelo coração em cada ciclo sistólico mas também da quantidade de luz absorvida por parte das artérias, veias, pele, entre outros.

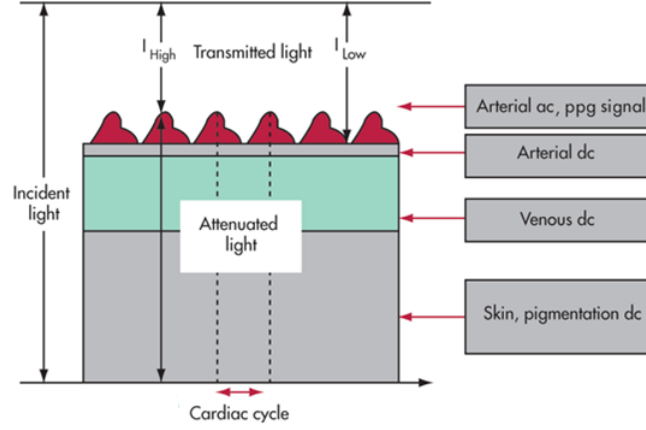


Figura 2.4: Variações na atenuação da luz pelos vários tecidos [Tex13].

Durante o ciclo sistólico, ou seja quando a pulsação atinge o seu pico, o volume de sangue nos tecidos aumenta. Esta quantidade extra de sangue e consequentemente de hemoglobina absorve mais luz, reduzindo a intensidade de luz que é transmitida/refletida para o fotodetector (I_{High} - figura 2.4). Por outro lado, durante o ciclo diastólico menos sangue está presente nas artérias, provocando um aumento da luz transmitida/refletida (I_{Low} - figura 2.4) [Joh97].

Desta forma, considera-se uma componente variável (arterial AC - figura 2.4) à parte oscilante do sinal PPG e como uma componente contínua (arterial DC - figura 2.4) ao *offset* que resulta da presença da pele, veias e outros tecidos. A dependência na intensidade de luz transmitida/refletida pode ser interpretada usando uma técnica de normalização onde a componente arterial AC é dividida pela componente arterial DC, dando origem a um rácio (R) sobre a absorção de luz. A equação 2.3 exprime essa dependência [Yit89], onde AC_R e DC_R correspondem à componente arterial AC e DC, respetivamente, relacionada com o LED (*light emitting diode*) de luz vermelha. De maneira semelhante, AC_{IR} e DC_{IR} correspondem à componente arterial AC e DC, respetivamente, associada ao LED de luz infravermelha [Yit89].

$$R = \frac{\frac{AC_R}{DC_R}}{\frac{AC_{IR}}{DC_{IR}}} \quad (2.3)$$

Desta maneira, a absorção de luz invariante no tempo devido à pele, ossos e outros tecidos está contemplada no novo método, contornando os problemas que a lei de Beer-Lambert apresenta.

2.2.3 Calibração

Duas equações independentes podem ainda ser derivadas a partir da equação 2.2 que descrevem a absorção de Hb e de HbO_2 . Resolvendo-as é possível achar uma equação relativa

ao S_pO_2 dependente da razão R (equação 2.4) [Yit89], onde A e B são coeficientes e R é o rácio de absorção de luz.

$$S_pO_2 = A + B \times R \quad (2.4)$$

As constantes A e B são achadas empiricamente através de amostras feitas a um grande número de indivíduos e devem ser correlacionadas com uma curva de calibração empírica, representada pela figura 2.5, para extração do S_pO_2 .

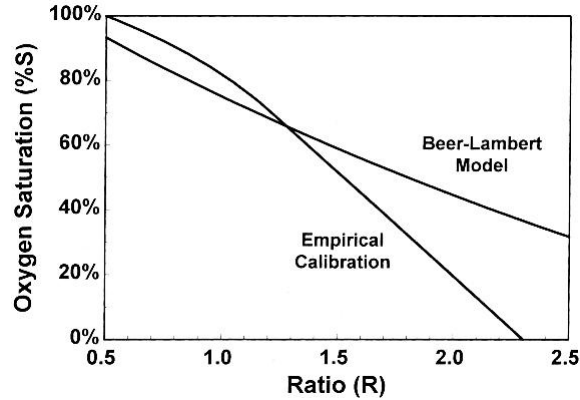


Figura 2.5: Relação empírica e modelo de Beer-Lambert entre o S_pO_2 e o rácio R [Des16].

Ao observar a figura 2.5 é possível reconhecer a quantidade de erro que seria introduzido num sistema que utilizasse a lei de Beer-Lambert para calibrar um oxímetro, em comparação com a calibração empírica (mais próxima da realidade).

2.2.4 Fontes de erro

Os oxímetros de pulso, usados em ambientes de monitorização em tempo real, deparam-se com algumas limitações que podem levar a leituras incorretas do S_pO_2 de um paciente. As mais relevantes são:

- **Movimento aplicado à sonda** - é um problema associado à distorção nas leituras do sinal PPG causado pelo movimento do paciente, levando a um deslocamento da sonda. Estes movimentos podem ter uma grande amplitude e consequentemente podem distorcer a pequena componente AC do sinal PPG recolhido ou até saturar o *front-end* do oxímetro [Joh97, SHR93].

A solução para este problema passaria pela incorporação de um acelerómetro no sistema para captar os movimentos do paciente e assim conseguir retirar os dados em momentos de menor movimento.

- **Luz ambiente** - este fator pode distorcer as leituras do sinal PPG levando a leituras incorretas do S_pO_2 , uma vez que esta luz não tem origem nos LEDs do oxímetro e portanto não deveria ser contabilizada. Para travar a deteção da luz ambiente os sensores de oximetria são normalmente isolados com um material opaco garantindo assim que nenhuma luz exterior chega ao fotodetector; poder-se-ia também pressionar firmemente

a sonda contra o tecido humano, embora não seja um meio de solucionar o problema tão eficaz como o anterior [Dre06].

- **Hipotensão, anemia e vasoconstrição** - quantidades baixas de glóbulos vermelhos (anemia) ou poucos vasos sanguíneos na zona onde a sonda está colocada podem originar erros no sinal PPG [SK90].

O mesmo se aplica a situações de vasoconstrição, quando por exemplo, prendemos voluntariamente a respiração ou em casos de hipotermia [LBB02].

- **Materiais que provocam obstrução de luz (e.g verniz)** - o uso de verniz nas unhas por parte de alguns pacientes impede a chegada correta de luz ao fotodetector, originando dados incorretos referentes ao sinal PPG [Wor11].

2.3 Oximetria de pulso

A oximetria de pulso é uma técnica que mede a quantidade de oxigénio no sangue de um paciente e indiretamente a sua frequência cardíaca. Esta técnica é muito conveniente pelo facto de ser não-invasiva, já que através da emissão de luz (LEDs) para o corpo humano e a receção de alguma dessa luz que é refletida, consegue-se extrair as fluações de sangue nas artérias [Joh97, Wor11]. Além disso, esta técnica pode ser facilmente integrada num *wearable device* (figura 2.6), daí a designação oxímetro de pulso, tornando a monitorização do ritmo cardíaco e S_pO_2 possível tanto para triagem de pacientes em ambiente hospital como nas suas habitações.



Figura 2.6: Oxímetro de pulso [Oxi16d].

2.3.1 Princípios de operação

Tradicionalmente, para haver um diagnóstico acerca do ritmo cardíaco, era usado um método manual analisando a pulsação no pulso ou na artéria do pescoço; é uma técnica com relativa precisão mas torna-se inapropriada quando se requer uma constante monitorização. Na mesma altura, para se determinar a quantidade de oxigénio presente no sangue, era usado um analisador de gases, no entanto é um método invasivo que provoca dor ao paciente já que é necessário a extração de uma amostra de sangue.

A oximetria de pulso vem inovar este domínio por ser uma técnica ótica não-invasiva para monitorização do ritmo cardíaco e S_pO_2 , que não requer contacto elétrico com o paciente e funciona com um baixo consumo de energia [Joh97].

Esta técnica, composta por um fotodetetor e por dois LEDs ativados alternadamente, aproveita bem o facto da Hb e HbO_2 terem uma absorção de espectro diferente. Estes LEDs devem ter comprimentos de onda onde a absorção por parte da Hb e HbO_2 sejam diferentes, normalmente utilizando um LED a 660 nm (luz vermelha) e um a 940 nm (luz infravermelha) [Joh97, Fox11].

O sensor que contém o fotodetetor e os dois LEDs é normalmente colocado em contacto com o dedo do paciente e através do processo retratado acima valores como o S_pO_2 e ritmo cardíaco podem ser obtidos. O fotodetetor pode ser utilizado de dois modos: oximetria de reflexão ou oximetria de transmissão, como se pode observar pela figura 2.7.

Nos oxímetros de transmissão os LEDs são colocados num lado da sonda, enquanto que o fotodetetor é colocado no lado oposto e voltado para os LEDs. O que se espera é uma emissão de luz, seguida de uma absorção de alguma dessa luz por parte dos tecidos humanos e a recolha da luz remanescente do outro lado por parte do fotodetetor [IB14].

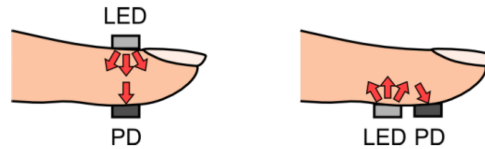


Figura 2.7: Sonda de transmissão (esquerda) e sonda de reflexão (direita) [TMSY14].

Por sua vez, a oximetria de reflexão coloca os LEDs e fotodetetor na mesma superfície da sonda, diferenciando apenas do método anterior na luz que é recolhida, já que neste caso, o fotodetetor recolhe a luz refletida pelos tecidos [IB14].

Comparando os dois tipos de sonda é importante referir que as sondas de reflexão têm menor SNR (*signal-to-noise ratio*) comparativamente com as sondas de transmissão, contudo são mais versáteis e podem ser utilizadas em mais zonas do corpo humano [Joh97].

2.3.2 Tipos de sondas e modelos comerciais

Devido à sua forma de funcionamento as sondas de reflexão podem ser usadas em qualquer parte do corpo humano como na têmpora, testa ou no esterno, enquanto que as sondas de transmissão são utilizadas basicamente nas extremidades do corpo humano como os dedos ou orelhas [IB14].

Sonda de transmissão para o dedo

As sondas de transmissão para os dedos (figura 2.8), presentes em qualquer ambiente hospitalar, são as mais utilizadas devido ao seu rápido uso no paciente e fácil remoção. Porém, apresentam algumas limitações como [PMD09]:

- Baixa resistência mecânica, já que a sonda após algum uso pode não conseguir a fixação ao dedo pretendida;
- Dificuldade no desenvolvimento de uma sonda que abranja os dedos tanto de crianças como de adultos;
- Baixa pressão sanguínea, afeta a precisão das medidas.



Figura 2.8: Sonda de transmissão para o dedo [Oxi16c].

Sonda de transmissão para a orelha

Ao contrário da sonda da figura 2.8, a sonda da figura 2.9 usa a orelha como ponto de extração de dados, indicada para situações de acidente onde há uma enorme perda de sangue (nos dedos ou mãos). São fáceis de usar e requerem pouco ajuste entre pacientes, no entanto são muito vulneráveis à vasoconstricção devido ao frio ou hipotermia, já que a circulação de sangue nesta extremidade do corpo é reduzida [PMD09].

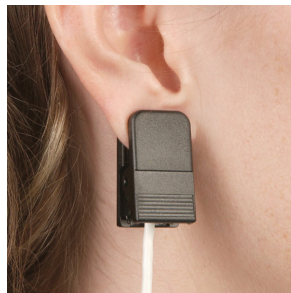


Figura 2.9: Sonda de transmissão para a orelha [Oxi16a].

Sonda de reflexão para a testa

As sondas de reflexão mais comuns são projetadas para a testa (figura 2.10), fornecendo uma maior resistência mecânica e influenciando assim a sua durabilidade.



Figura 2.10: Sonda de transmissão para a testa [Mas16].

Estas sondas são muito úteis quando um paciente possui uma pulsação fraca, uma vez que o sangue bombeado do coração chega à cabeça mais rapidamente do que aos dedos, permitindo a extração de resultados com melhor qualidade. Para além disso, não apresentam nenhum problema no uso em adultos e crianças [Nel03].

Em termos de desvantagens estas sondas possuem uma menor precisão nas leituras e é necessário o ajuste deste sensor de paciente para paciente [PMD09].

Oxímetros no mercado

Impulsionado pela crescente necessidade dos oxímetros de pulso desempenharem funções de monitorização contínua de pacientes quer em ambiente hospitalar quer no domicílio, o mercado global para este tipo de dispositivos deverá alcançar os 2.3 mil milhões de dólares em 2020 [Mar16]. O aumento da procura destes dispositivos traduz-se numa maior concorrência entre as empresas, motivando desenvolvimentos ainda maiores em aparelhos de oximetria e consequentemente introduzindo no mercado novos oxímetros de pulso.

A tabela 2.1 fornece informações acerca de alguns oxímetros disponíveis no mercado em termos de precisão, custo e bateria necessária.

Tabela 2.1: Oxímetros disponíveis no mercado [NHS10].

Modelo	Bateria	Precisão	Custo
Avant 4000	7.2V recarregável	$\pm 2\%$ S_pO_2 $\pm 3\%$ Pulsação	1650\$
Nonin WristOx	2 x 1.5V N-cell	$\pm 2\%$ S_pO_2 $\pm 3\%$ Pulsação	725\$
SPO 7500	3.6V Lithium	$\pm 2\%$ S_pO_2 $\pm 3\%$ Pulsação	499\$
Daray V501	2 x AAA	$\pm 2\%$ S_pO_2 ± 2 BPM	195\$
ihealth PO3	3.7V Li-ion	$\pm 2\%$ S_pO_2 ± 2 BPM	151\$
Viamed VM-2101	2 x AAA	$\pm 2\%$ S_pO_2 ± 1 BPM	133\$
Rossmax SB-220	2 x AAA	$\pm 2\%$ S_pO_2 ± 3 BPM	94\$
Edan H10	2 x AAA	$\pm 3\%$ S_pO_2 ± 2 BPM	77\$
Beijing MD300	2 x AAA	$\pm 2\%$ S_pO_2 ± 2 BPM	65\$

Com a introdução do *Bluetooth* 4.0 é mais simples a criação de novos dispositivos com conexão ao *smartphone* e com um menor consumo de energia. O protocolo BLE (*Bluetooth low energy*) é perfeito para transmitir pequenos pacotes de dados relativos a movimentos, sinais vitais ou parâmetros *fitness* que se pretendam exibir no *smartphone* ou *smartwatch*. Por essa razão é que estão disponíveis no mercado atual oxímetros de pulso incorporando *Bluetooth*. Estes novos dispositivos permitem uma monitorização remota dos pacientes, motivando cada

vez mais avanços científicos nesta área [NXP14], como é o caso do oxímetro de pulso com ligação BLE e ecrã de visualização de resultados da figura 2.11.



Figura 2.11: Oxímetro de pulso atual [Oxi16b].

Neste capítulo foram apresentados temas como a hemoglobina, S_pO_2 e oximetria de pulso. No capítulo seguinte será realizada uma apresentação sobre o papel dos *wearable devices* na IoT, que plataformas móveis existem, como são utilizadas e quais são os serviços de saúde à distância existentes.

Capítulo 3

Dispositivos móveis

As redes móveis atuais fornecem uma enorme conectividade entre dispositivos, permitindo o desenvolvimento de serviços e aplicações inovadoras. Esta grande capacidade de conectividade entre objetos está a propagar-se para além dos *laptops*, *tabletes*, *smartphones*; para carros, *wearable devices*, edifícios, sensores, televisões, consolas de jogos, entre outros, originando a formação da IoT.

3.1 *Internet of things*

A IoT é um enorme sistema de dispositivos interligados através da *Internet*. Cada um destes dispositivos tem um identificador único e a capacidade de transferir dados para a rede sem a necessidade da interação humana [Eva11]. Um exemplo simples de como definir a IoT (figura 3.1) seria um cenário em que a nossa despensa nos alertasse que já não resta muita quantidade de um certo alimento e que é tempo de repô-la.

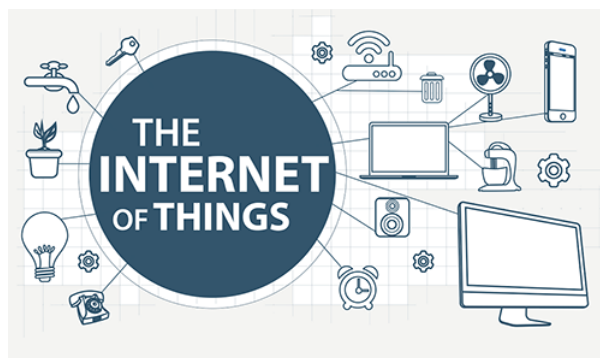


Figura 3.1: *Internet of Things* [oT16].

Tem havido uma distribuição maciça de dispositivos no mercado atual, capazes de resolver problemas em todos os setores. Cada um destes dispositivos contém vários sensores que comunicam com outros através de uma rede nodal, tornando as tarefas cada vez mais simples. É esperado que um veículo recente possua cerca de 80 sensores. Não só os veículos mas também as habitações, *smartphones* e aparelhos eletrónicos estão cada vez mais inteligentes, originando a passagem para uma era de conectividade a todos os níveis [Shr15] .

3.1.1 *Wearable devices*

Wearable technology é um termo usado para descrever os mais diversos tipos de dispositivos eletrônicos que podem ser usados em qualquer indivíduo de forma tão simples como uma peça de roupa. Estes dispositivos recolhem e fornecem informação ao utilizador sobre algum parâmetro, como é o caso dos dispositivos de monitorização (figura 3.2) ou a roupa inteligente [PWC14].

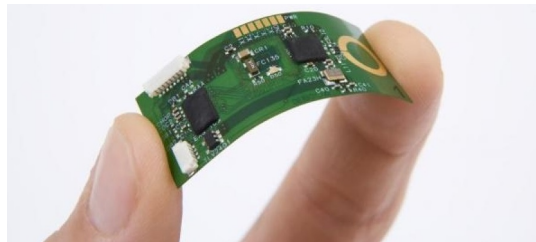


Figura 3.2: Sensor flexível de ECG com comunicação *wireless* [Sci16].

Nos dias de hoje, com o aumento da quantidade dos *wearable devices*, é possível analisar dados em larga escala de uma forma simples. Muitos dispositivos tem sido desenvolvidos ao longo dos últimos anos para recolherem dados do corpo humano de uma maneira portátil e adequada a cada situação como a frequência cardíaca, S_pO_2 , ECG, pressão sanguínea ou temperatura corporal, tornando a saúde uma das mais importantes áreas de aplicação destes dispositivos. Em ambientes hospitalares e à medida que os componentes de *hardware* sofrem uma diminuição no seu tamanho, estes dispositivos de recolha de dados podem ser colocados em contacto com a pele, criando uma unidade de monitorização eficiente e compacta. Estes avanços possibilitam a realização de atendimentos de saúde à distância, proporcionando uma experiência mais confortável ao paciente [Kho15].

Outra área de aplicação importante dos *wearable devices* são os dispositivos para *fitness*. Estes dispositivos são normalmente compostos por um pequeno circuito de *hardware* (e.g pulseira) e por um *software* (aplicação móvel) para analisar e processar os dados recolhidos, tal como apresentado pela figura 3.3 [SMR14].



Figura 3.3: Dispositivo e aplicação *fitness* [Dan14].

Existem vários *wearable devices* no mercado de *fitness* atual (figura 3.4) incorporando acelerómetros, giroscópios ou GPS (*global positioning system*) para estarem aptos a recolherem

informações relativas ao ritmo cardíaco, passos, energia gasta e distância percorrida qualquer utilizador [SMR14].



Figura 3.4: Conjuntos de *wearable devices* [Dev16].

Os dispositivos exibidos na figura 3.4 ajudam a população, nomeadamente desportistas, a controlar de uma forma mais eficiente a sua condição física e consequentemente a sua condição clínica, resultando numa melhor qualidade de vida.

3.1.2 Serviços de saúde

Tem havido um enorme crescimento na utilização e criação de *wearable devices* com funções de monitorização, nomeadamente em sistemas ECG portáteis para pacientes de alto risco. O sistema móvel de monitorização de pacientes é uma das técnicas que reduz os custos associados aos cuidados médicos tradicionais. Estes sistemas, como é o caso da figura 3.5, adquirem sinais vitais e transmitem-os através de uma rede de telecomunicações para um dispositivo onde os especialistas de saúde podem verificar o estado clínico do paciente, detetando eventuais anomalias [PPB⁺12].

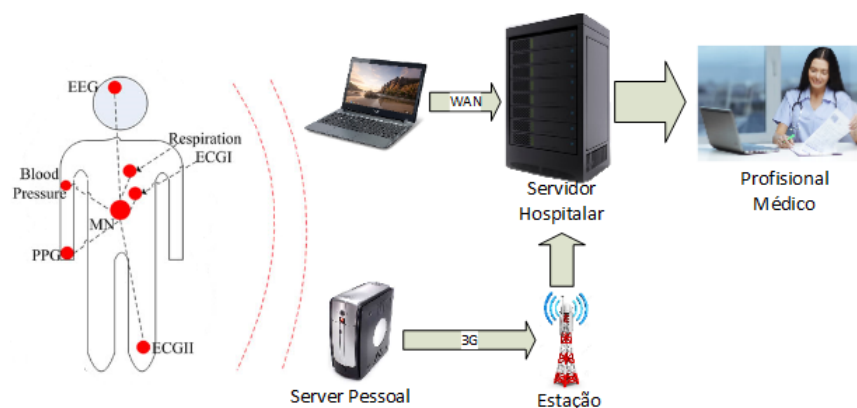


Figura 3.5: Sistema remoto de saúde genérico.

Apesar dos inúmeros benefícios concedidos pelos sistemas de monitorização, a utilização destes sistemas a nível global é ainda reduzida [Kay11]. Alguns destes sistemas são resumidos de seguida:

- **Intelligent Mobile Health Monitoring System (IMHMS)** - este sistema prevê de forma inteligente os estados de saúde do paciente e fornecer informação através de dispositivos móveis. O IMHMS usa *Wearable Wireless Body Area Network* para coletar os dados relativos a cada paciente, armazenando-os numa base de dados. Este sistema fornece ao utilizador flexibilidade, simplicidade e uma interface *user-friendly* [SBKA09].
- **MobiHealthcare System (MHCS)** - o MHCS é constituído por sensores que colem vários sinais vitais. O sistema é desenvolvido num servidor que possui uma grande capacidade de armazenamento de dados e onde se pode visualizar o estado clínico de cada paciente [MMSL12].
- **Wireless Intelligent Sensor System (WISS)** - este sistema foi desenvolvido com o intuito de monitorizar pacientes em tempo real utilizando sensores *plug-and-play*. O sistema prevê aspetos como o stress de soldados sobre condições extremas, fazendo uso de comunicações de curto alcance para reduzir o consumo de energia e aumentar a segurança [JLR⁺03].
- **CardioSentinal (CS)** - o CS, apresentado na figura 3.6, tem como objetivo fornecer um serviço de controlo para a população envelhecida 24h por dia. É constituído por sensores, comunicação de curto alcance, computação ubíqua, redes móveis e centros de dados [GZN⁺12].

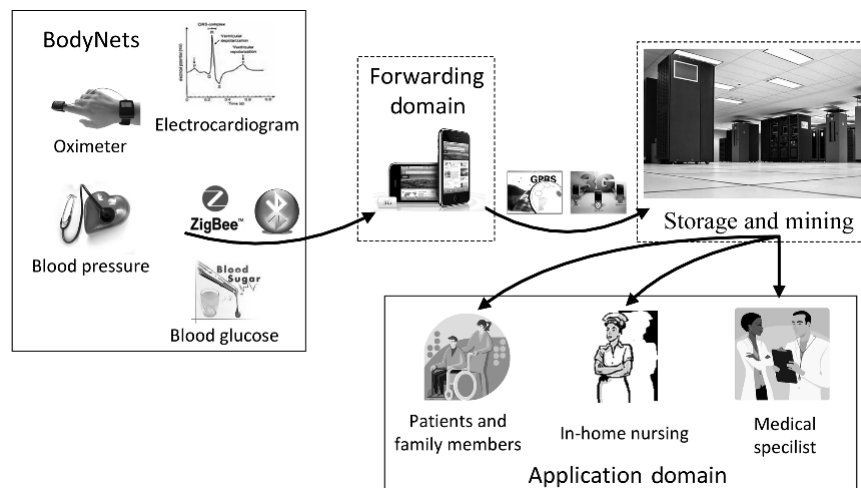


Figura 3.6: Arquitetura do CardioSentinal [Ols08].

- **Personalized Heart monitoring (PHM)** - este sistema usa sensores sem fios e *smartphones* para monitorizar o estado de saúde de pacientes com graves problemas cardíacos. O *smartphone* tem a função de detetar quedas, examinar em tempo real os dados do traçado ECG, localizar o paciente e determinar se necessita de apoio médico. A única desvantagem do sistema é o facto do *smartphone* possuir uma bateria com baixa

autonomia, durando apenas oito horas quando conectada continuamente ao dispositivo ECG [GL07].

Para além de sistemas desta natureza, é possível desenvolver sistemas de baixo custo com dispositivos e tecnologias de uso geral como os *smartphones* e *Bluetooth*, que podem ser usados para a monitorização pessoal de vários sinais vitais.

3.2 Smartphones

A computação móvel ganhou uma enorme relevância nas últimas décadas. Os *smartphones* são hoje em dia um dos aparelhos mais usados pela população, possuindo um grande potencial na área da monitorização, devido à [NB13]:

- Dimensão cada vez mais reduzida dos componentes de *hardware*, tornando os telemóveis similares a minicomputadores (*smarthpones*).
- IoT, possibilitando um controlo remoto de todos os dispositivos através da rede.

Não há muito tempo os dispositivos móveis possuíam tamanhos volumosos e a conectividade à rede possuía maior latência, menor largura de banda e pior cobertura, porém, nos dias correntes com a explosão da *internet* os *smartphones* fornecem ao utilizador acesso aos *media*, *e-mail*, *banking* ou redes sociais. Estas ferramentas devem-se ao facto de além de dispositivos móveis, os *smartphones* são incorporados com um sistema robusto, grande capacidade de processamento de dados, conectividade à *internet*, câmara, GPS e microfone, fazendo com que este dispositivo tenha vindo a ser alvo de vários estudos relacionados com a monitorização médica à distância [QG12].

A última década levou ao crescimento de várias plataformas móveis como a *IOS*, *Android* e *Windows Phone*, porém, devido ao enorme crescimento da quota de mercado da *Google Android*, esta tornou-se a líder de mercado atual com cerca de 88% da quota total [mS16].

3.2.1 Android

O *Android* é um sistema operativo desenvolvido pela *Google*, *open-source* e presente no mercado desde 2005. As aplicações *Android* são normalmente escritas em *Java* e a interface de utilizador (UI) é normalmente construída com o uso do *Extensible Markup Language* (XML), que fornece uma interface gráfica para gerir objetos no ecrã do *smartphone* como ilustrado na figura 3.7.

Para além disso, com a explosão no mercado dos *tablets*, a *Google* melhorou o seu sistema operativo com as gerações 4.0, 5.0 e 6.0, fornecendo a possibilidade de se desenvolver aplicações tanto para *smartphones* como para *tablets*. Para tal, o ambiente de desenvolvimento disponível para *Windows*, *Linux* e *Mac OS X* assenta no *Android Software Development Kit* (SDK), que fornece as ferramentas necessárias para o desenvolvimento de qualquer aplicação.

Os dispositivos *Android* integram sensores como acelerómetro, giroscópio, bússola, GPS, comunicações *Bluetooth*, GSM (*global system for mobile communication*) e *Wi-Fi*, tornando-os versáteis e compatíveis com uma grande quantidade de *hardware*.

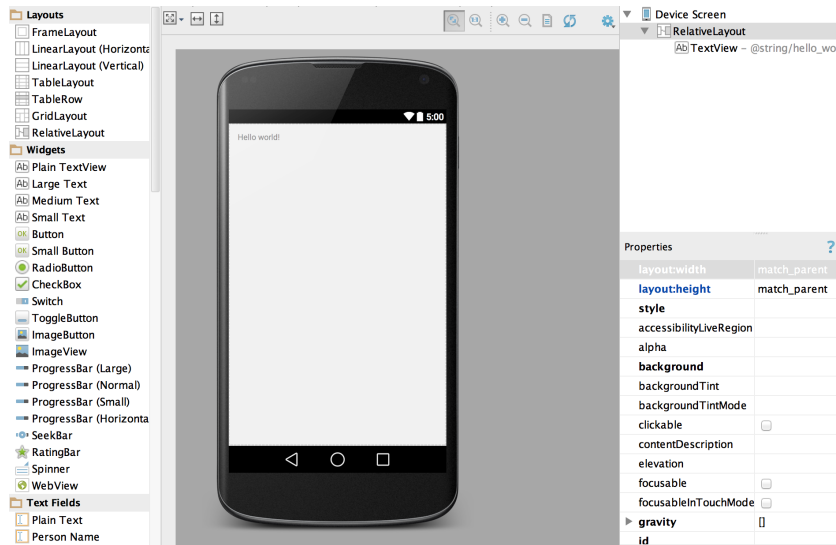


Figura 3.7: Interface gráfica para desenvolvimento das aplicações *Android*.

3.2.2 *Bluetooth*

O *Bluetooth* é uma tecnologia de rádio sem fios que opera a 2.4 GHz projetada para atender necessidades específicas de envio e receção de dados a curtas distâncias. O desenvolvimento desta tecnologia começou em 1994 quando Ericsson procurava uma maneira de estabelecer uma conexão sem fios entre um teclado e um computador. Depois de um estudo de viabilidade de curto alcance e baixa potência, esta ligação sem fios acabou por ser útil para muitos outros projetos [Blu16, Hew10].

Uma enorme variedade de produtos que incorporam esta tecnologia está hoje em dia disponível no mercado, como *laptops*, *smartphones*, impressoras, teclados e viaturas. De acordo com o relatório anual da Bluetooth SIG e como se pode constatar pela figura 3.8, desde 2012 o número anual de dispositivos *Bluetooth* tem vindo a aumentar e continuará até 2018, onde se prevê a existência de 4.9 mil milhões de aparelhos com esta tecnologia.

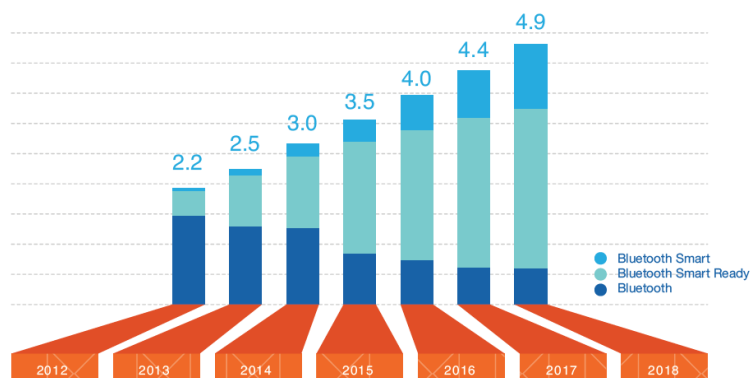


Figura 3.8: Produção de dispositivos *Bluetooth* no planeta (mil milhões) [SIG14].

É importante mencionar que os dispositivos com *Bluetooth smart* estão a ganhar cada vez mais mercado comparativamente ao *Bluetooth* clássico, isto deve-se ao desenvolvimento da IoT e à adoção cada vez mais acentuada do *Bluetooth smart* nos desportos, *fitness* e cenários médicos [SIG14].

O *Bluetooth* clássico pode consumir rapidamente a bateria de um projeto, uma vez que foi desenvolvido para trocar uma grande quantidade de dados num curto espaço de tempo. Por outro lado, o *Bluetooth smart* destina-se a fornecer um consumo energético consideravelmente reduzido.

Bluetooth low energy

O BLE (*Bluetooth low energy*) é uma subdivisão do *Bluetooth smart*. Devido ao seu baixo consumo de energia, o BLE não suporta *streaming* de dados. Por essa razão, após uma ligação ser estabelecida, este dispositivo passa a maior parte do tempo no modo de suspensão à espera de enviar/receber o próximo conjunto de informações; com uma velocidade de dados de 1 Mbps, permite uma rápida transferência de dados em pequenos pacotes. Uma vez transferidos os dados, o BLE volta a entrar num estado de consumo reduzido para economizar bateria, por sua vez, um dispositivo *Bluetooth* clássico permaneceria sempre ligado, independentemente dos dados transmitidos/recebidos [Blu16].

A contextualização dos dispositivos móveis com serviços de saúde à distância conclui o estado da arte desta dissertação. O capítulo seguinte especificará as características impostas ao sistema a desenvolver, como deve funcionar e como foi planeado em termos dos blocos necessários.

Capítulo 4

Especificação e arquitetura do sistema

Neste capítulo são abordados temas relativos aos requisitos impostos ao sistema e que blocos deve possuir para a criação de um sistema portátil de aquisição de sinais vitais.

4.1 Introdução

Um sistema que permita uma constante monitorização dos sinais vitais de um utilizador poderá ajudar a população a despistar com mais antecedência situações clínicas de risco. É com este propósito que se pretende desenvolver um sistema não-invasivo de monitorização de alguns sinais vitais, permitindo que a população tenha uma constante vigilância do seu estado de saúde. Os sinais avaliados pelo sistema serão a frequência cardíaca, S_pO_2 e temperatura corporal, facilitando o controlo do estado de saúde de qualquer indivíduo em qualquer lugar.

O funcionamento do sistema requererá um dispositivo de medição dos parâmetros vitais mas também o *smartphone* do utilizador para a consulta em tempo real dos resultados. Depois de acionado o botão de inicialização do sistema, será esperado que nos momentos seguintes se coloque o dedo em contacto com os sensores de medição. É importante manter o dedo o mais estável quanto possível para aumentar a fiabilidade da informação recolhida. Por fim, através da aplicação móvel do projeto, será possível consultar os resultados acerca da frequência cardíaca, S_pO_2 e temperatura, no *smartphone* do utilizador.

4.2 Requisitos

Este trabalho tem como propósito o desenvolvimento de um sistema modular capaz de fornecer informação acerca do ritmo cardíaco, S_pO_2 e temperatura. Este sistema deverá ser de fácil/rápida utilização e portátil. Necessitará de uma interface de utilização intuitiva para que pacientes mais idosos consigam monitorizar sem qualquer problema o seu estado de saúde. Além disso, deverá ser um dispositivo portátil para o que o utilizador consiga monitorizar em qualquer lugar os seus sinais vitais.

A oximetria de pulso de reflexão permite que as medições sejam realizadas a partir de áreas do corpo em que a oximetria de pulso de transmissão não pode ser aplicada, facilitando a sua utilização. Tendo isto em consideração, é necessário que o oxímetro escolhido seja de reflexão e capaz de recolher amostras com uma taxa de amostragem adequada para a largura

de banda do sinal cardíaco (rondando os 3 Hz), presente no pulso arterial. Por outro lado, como as variações significativas de temperatura acontecem aproximadamente de segundo em segundo, será necessária uma taxa de amostragem por parte do sensor de temperatura de 1Hz e que utilize tecnologia de infravermelhos, com o propósito de se obter a melhor precisão possível.

Com vista a portabilidade do dispositivo, necessitar-se-á de ter em consideração que todo o sistema deve ser alimentado entre 3.3V e 5V, uma vez que são os valores padrão usados pela maioria dos módulos e baterias. A sonda de oximetria deverá ser ainda conectada a uma unidade central, como um microcontrolador, que seja capaz de adquirir e processar os sinais provenientes dos sensores de oximetria e temperatura.

Apesar de existirem ferramentas para analisar o ritmo cardíaco e S_pO_2 em *wearable devices*, existe ainda uma carência de soluções com *hardware standard*, portáteis, de baixo custo, com protocolos de comunicação *standard*, formato de dados abertos e extensíveis. Desta forma, surge a necessidade de criar um sistema de aquisição de dados vitais com estas características.

4.3 Arquitetura do sistema

O primeiro passo no desenvolvimento deste projeto passou pela criação de um esboço da arquitetura do sistema. Foram analisados os vários blocos que seriam necessários e a forma como se interligariam uns aos outros para ser possível a aquisição e processamento de toda a informação, desde o sensor até ao monitor de visualização de resultados. A figura 4.1 apresenta um esquema geral da arquitetura proposta para o sistema de monitorização de sinais vitais com ligação a um dispositivo móvel.

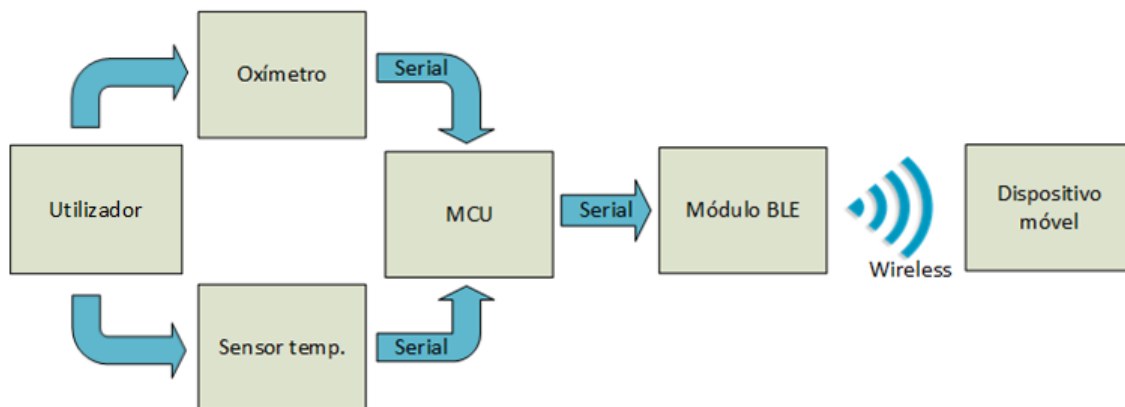


Figura 4.1: Diagrama de blocos do sistema.

A solução proposta para satisfazer os requisitos do projeto é composta por um sensor de oximetria, responsável pela recolha de amostras relativas às palpitações de sangue nas artérias, e por um sensor de temperatura de infravermelhos. Estas amostras serão armazenadas pelos próprios módulos e posteriormente enviadas para o MCU (*microcontroller unit*) do sistema através de uma comunicação série. Na fase seguinte, o MCU será responsável pela construção dos pacotes de dados que posteriormente serão enviados para o dispositivo móvel. O envio dos dados para o dispositivo móvel será assegurado por um módulo BLE, que receberá os dados via série do MCU e transmiti-los-á através de uma conexão *wireless* para um dispositivo

móvel. Por fim, depois de algum tratamento dos pacotes recebidos, será possível observar em tempo real, no dispositivo móvel, valores instantâneos do ritmo cardíaco, S_pO_2 e temperatura do utilizador, assim como um gráfico da sua atividade cardíaca.

Seguindo o diagrama da figura 4.1 é feita de seguida uma breve descrição acerca das características, funcionalidades e dados transferidos e recebidos por cada bloco.

4.3.1 Oxímetro

Na escolha do módulo de oximetria será essencial encontrar um módulo de aquisição de dados de dimensões reduzidas e que permita ser facilmente acoplado a um microcontrolador. Este bloco será responsável pela extração de valores relativos à quantidade de luz refletida pela hemoglobina presente no sangue do utilizador, logo, terá de atuar sobre os LEDs vermelho e infravermelho, condicionar o sinal analógico obtido (através de filtros e amplificadores) e armazenar numa estrutura interna a informação recebida.

O bloco deverá ser capaz de transferir posteriormente os valores ADC (*analog-to-digital conversion*) de cada amostra para um MCU através de uma comunicação série genérica, como I^2C (*inter-integrated circuit*), SPI (*serial peripheral interface*) ou RS232.

4.3.2 Sensor de temperatura

Um dos mais importantes sinais vitais é a temperatura corporal que varia dependendo da zona do corpo em que é realizada a medição, da hora do dia e da atividade física. Além disso, a temperatura também varia dependendo do sono, frio e fome. Não há um valor exato que a temperatura corporal deve ter, no entanto, é comum em média a temperatura de um humano situar-se entre 37 e 37.7 °C.

A medição de temperatura sem contato é baseada na propriedade de que todos os materiais emitem radiação eletromagnética (radiação infravermelha). Estes materiais emitem radiação infravermelha proporcional à sua temperatura, posteriormente recolhida pelo detector. Por fim é realizada uma diferença entre a radiação recebida e a temperatura original do detetor, para assim ser extraída a temperatura do corpo que se pretende medir.

O bloco de medição da temperatura deverá estar apto a recolher valores da temperatura corporal instantânea com uma casa decimal de resolução e com tecnologia de infravermelhos. Posteriormente, enviará estes valores para o MCU através de uma interface série I^2C , SPI ou RS232.

4.3.3 Microcontrolador

Qualquer sistema precisa de incorporar uma unidade central para processar os dados e atuar como *master*. Um MCU difere de um processador de uso geral, usado em típicos PC's, por possuir dimensões reduzidas, controladores de comunicações, periféricos integrados, vários pinos I/O (*input/output*) e uma memória com capacidade mais limitada.

Este bloco será responsável, cada vez que o sistema é ligado, pela configuração inicial dos registos dos dois sensores utilizados. Além disso, terá que recolher os dados provenientes quer do sensor de temperatura quer do oxímetro e enviar posteriormente esses valores para um módulo BLE, pelo que necessitará de possuir interfaces de comunicação série como I^2C , SPI ou RS232.

4.3.4 Módulo BLE

Com a tecnologia atual, a implementação de um sistema sem fios que consiga trocar informação é cada vez mais simples. Existem essencialmente três tecnologias *low-power* sem fios: *Wi-Fi*, *ZigBee* e *Bluetooth*.

Uma vez que um dos requisitos impostos a este sistema de aquisição de dados vitais é que possa ser usado no exterior e apesar de nos últimos anos terem sido desenvolvidos módulos *Wi-Fi* com consumos mais reduzidos relativamente ao *ZigBee* e *Bluetooth*, a tecnologia *Wi-Fi* não pode ser aplicada a este sistema por requerer uma conexão à *internet*. Por outro lado, o *ZigBee* é um protocolo essencialmente para uso numa LAN (*local area network*) com vários *nodes*, já o BLE foi desenvolvido para ser usado numa rede pessoal com um baixo consumo. Por estas razões optou-se pela escolha de um módulo BLE para assegurar o envio de dados para o dispositivo móvel.

Este bloco assegurará a conversão e o posterior envio dos dados relativos à hemoglobina e à temperatura do utilizador para o dispositivo móvel, por isso, deverá ser capaz de receber os dados provenientes do MCU através de uma interface série I^2C , SPI ou RS232.

Finalizada a descrição preliminar de como é suposto o sistema funcionar e dos blocos necessários, no próximo capítulo é abordado o trabalho em termos da implementação dos vários módulos, recolha e transmissão de informação.

Capítulo 5

Implementação do módulo de sensores

Neste capítulo será apresentado todo o procedimento realizado para a aquisição dos dados, nomeadamente a implementação do módulo de sensores, como são realizadas as comunicações entre eles, como se configuram e como é recolhida a informação.

Inicialmente foi realizado um protótipo do sistema com o microcontrolador CC1110 mas com a evolução do projeto, sentiu-se a necessidade de o substituir por um Arduino.

5.1 Sistema proposto

O sistema projetado (figura 5.1) é composto por dois sensores: um oxímetro, responsável pela aquisição de amostras relativas à luz refletida pela hemoglobina e por um sensor de temperatura. Estes dois sensores recolhem amostras de uma forma não-invasiva, que mais tarde são transferidas para o CC1110/Arduino (MCU). Nesta unidade são construídos os pacotes de dados e enviados através de uma comunicação RS232 para o módulo BLE, que por sua vez converte os dados recebidos em formato *Bluetooth low energy*. No final, encontra-se um *smartphone Android* preparado para recolher os pacotes de dados, processar a informação recebida e exibir no ecrã os resultados em tempo real.

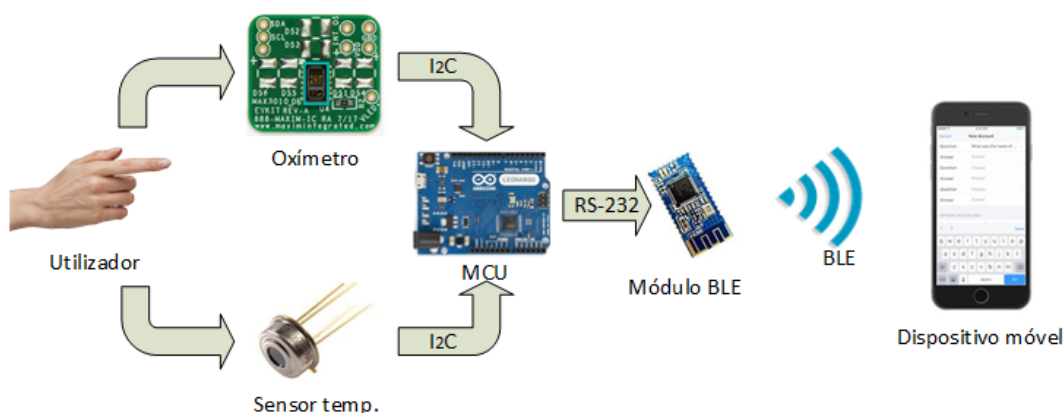


Figura 5.1: Diagrama do sistema proposto.

5.1.1 Oxímetro - MAX30102

A Maxim introduziu no mercado o MAX30102, um oxímetro de pulso direcionado para dispositivos portáteis, aplicações médicas e *fitness*. Este *chip*, caracterizado como *ultra-low power* devido aos seus consumos abaixo de 10 mW, permite ao sistema dispor de uma boa autonomia.

O MAX30102 possui um FIFO circular onde armazena as amostras dos LEDs vermelho e infravermelho relativas às variações de hemoglobina, e como pode ser visto na figura 5.2, este módulo integra todos os blocos de acondicionamento de sinal (filtros e amplificadores) necessários e utiliza uma interface de comunicação I^2C para enviar os dados para um *master*. Além disso possui um pino adicional, ativado através de uma interrupção, para alertar quando está disponível uma nova amostra no FIFO.

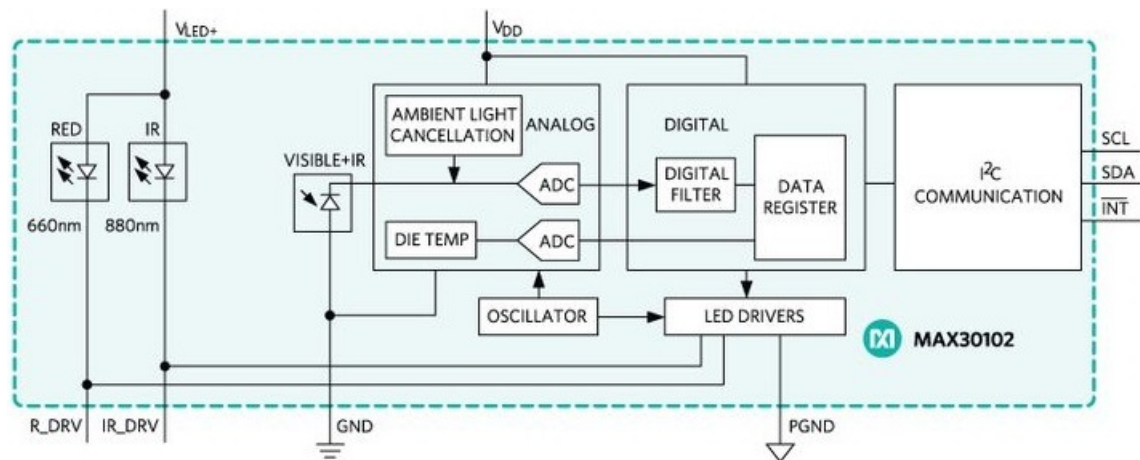


Figura 5.2: Diagrama do MAX30102 [Max15].

5.1.2 Sensor de temperatura - MLX90614

O MLX90614, o sensor de temperatura de infravermelhos escolhido para desenvolver o projeto, é indicado para monitorizar a temperatura de um objeto sem contato, nomeadamente para a medição da temperatura corporal. O sensor é alimentado a 3.3V, fornece uma resolução de 0.02 °C e possui uma interface de comunicação I^2C .

5.1.3 Microcontroladores

A execução do projeto levou ao desenvolvimento de duas provas de conceito com diferentes microcontroladores. Na fase inicial foi utilizado o microcontrolador CC1110, posteriormente substituído por um Arduino.

CC1110

O CC1110 é um *chip ultra-low power* com comunicação rádio, composto pelo MCU 8051. Apresenta consumos abaixo de 10 mA, incorpora 21 GPIOs (*general-purpose input/output*), duas UARTs (*universal asynchronous receiver/transmitter*) e quatro *timers*, tornando possível

a integração dos blocos do projeto. A placa de desenvolvimento SmartRF04EB foi utilizada já que contém a lógica necessária para o utilizador se familiarizar com o MCU escolhido e aprender a programá-lo. Esta placa contém um LCD (*liquid crystal display*), USB (*universal serial bus*), LEDs, botões e interfaces simplificadas para todos os elementos. O IDE (ambiente de desenvolvimento integrado) escolhido para desenvolver o *software* no CC1110 foi o IAR Embedded Workbench uma vez que é compatível com este microcontrolador, permitindo desenvolver, compilar e fazer *debug* a programas em linguagem assembler e C/C++.

Como a ideia inicial era incluir no projeto um sensor de tensão arterial, um sensor de oximetria, um sensor de temperatura, pensou-se em integrar cada um destes sensores com um CC1110 com a intenção do utilizador poder colocar cada sensor na zona do corpo mais adequada. A ideia seria criar uma rede via rádio (quatro *nodes*), na qual cada um dos sensores (*end-devices*) enviaria os seus dados para um *access-point*, utilizando o protocolo rádio SimpliciTI. Desta forma, cada *end-device* recolhia dados acerca dos vários sinais vitais e o *access-point* transmitiria esses pacotes de informação para o *smartphone*, através de uma *gateway* BLE.

Arduino

O Arduino Leonardo é composto pelo ATmega32U4, 23 pinos de *input/output* (digitais, analógicos e pulse-width modulation), uma conexão micro USB para o PC e possui a lógica necessária para o utilizador desenvolver qualquer projeto, tornando-o numa excelente escolha na integração dos módulos usados. O IDE *open-source* fornecido pelo Arduino foi utilizado para o desenvolvimento de todo o *software* deste microcontrolador.

Como já foi referido, a ideia inicial seria do sistema incluir um módulo de tensão arterial, para fornecer uma informação mais completa do estado clínico do utilizador. Porém, como não foi possível em tempo útil a aquisição de um módulo aberto e a custos razoáveis, deixou de fazer sentido ter a arquitetura apresentada na secção 5.1.3 - CC1110. Por essas razões foi desenvolvida uma versão mais económica, na qual os sensores de oximetria e temperatura enviam os dados para um Arduino Leonardo, onde são criados pacotes de dados para serem transferidos para um *smartphone* através de um módulo BLE.

5.1.4 Módulo BLE - Bluefruit LE UART

O Bluefruit LE UART, colocado no mercado pela Adafruit, foi o módulo BLE escolhido para estabelecer a comunicação entre o MCU e o dispositivo móvel *Android*. Esta empresa disponibiliza uma aplicação móvel para teste e bibliotecas de código, tornando mais fácil incorporar conectividade BLE a um projeto.

Alimentado a 3.3V e com um consumo de corrente de pico inferior a 20 mA, este módulo recebe os dados do MCU através de uma interface UART e converte-os para *Bluetooth low energy*. Para isso é necessário conectar o seu pino de receção a um porto do MCU que seja capaz de transmitir informação. Por sua vez, o pino de transmissão deste módulo é usado para enviar dados para o MCU. O Bluefruit LE UART tem ainda dois pinos adicionais, CTS (*clear to send*) e RTS (*read to send*), responsáveis pelo controlo de fluxo de dados.

5.2 Interligação entre MCU, sensores e módulo BLE

A comunicação entre os sensores e o MCU é realizada através de I^2C , enquanto que a conexão entre o MCU e o módulo BLE é assegurada por RS-232 (figura 5.3).

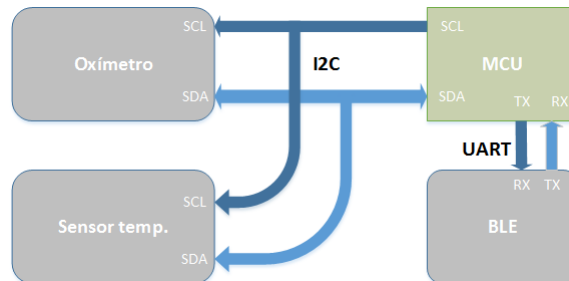


Figura 5.3: Interfaces de comunicação.

Projetado essencialmente para o controlo de ICs (circuitos integrados) simples, o I^2C é um protocolo de comunicação série bidirecional, composto por um barramento com uma linha de dados (SDA) e uma linha com o sinal de relógio (SCL).

5.2.1 Oxímetro

O MAX30102 está equipado com um barramento de 2 linhas (SDA e SCL), compatível com o protocolo I^2C . O MCU deste projeto (*master*) terá de gerar o sinal SCL e inicializar a transferência de dados pelo barramento.

O estado das linhas SDA e SCL devem manter-se em *high* quando o barramento não estiver a ser usado. No entanto quando o *master* pretender inicializar uma comunicação, deve enviar uma sequência de *start*. Este comando traduz-se numa transição de *high* para *low* na linha SDA com a linha SCL no estado *high*. Por outro lado, quando a ligação ao MAX30102 já não for necessária deve-se enviar uma sequência de *stop*, que consiste numa transição de *low* para *high* na linha SDA com a linha SCL no estado *high*.

Depois da condição de *start*, o *master* deve enviar o endereço do MAX30102. Este endereço é composto por um ID (*identification*) de 7 bits e um LSb (bit menos significativo) bit que indica se o *master* pretende ler ou escrever dados no MAX30102. A informação necessária acerca das configurações de fábrica deste elemento pode ser consultada de seguida:

```
MAX30102 Address 0b1010111x
Leitura LSb = 1
Escrita LSb = 0
```

Por fim, o MAX30102 utiliza, como mecanismo usual de I^2C , um nono bit (*acknowledge*) para confirmar que os últimos dados foram recebidos. O MAX30102 coloca a linha SDA no estado *low* durante a transferência deste bit se a informação precedente tiver sido recebida com sucesso. Numa operação de leitura, o *master* (MCU) coloca a linha SDA no estado *low* durante o nono bit para prosseguir com a leitura de dados. Colocar o estado da linha SDA em *high* durante a passagem deste bit significa que o *master* leu o último byte de dados desejado do MAX30102.

Tendo estas informações em consideração, uma operação de transferência de dados para o MAX30102 seguirá o esquema da figura 5.4.

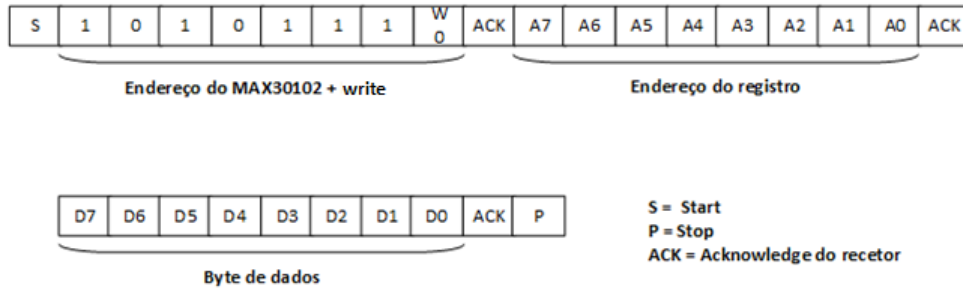


Figura 5.4: Configuração de um registo do MAX30102

Para efetuar uma leitura de dados no MAX30102 são necessárias duas operações I^2C , como se pode ver pela figura 5.5.

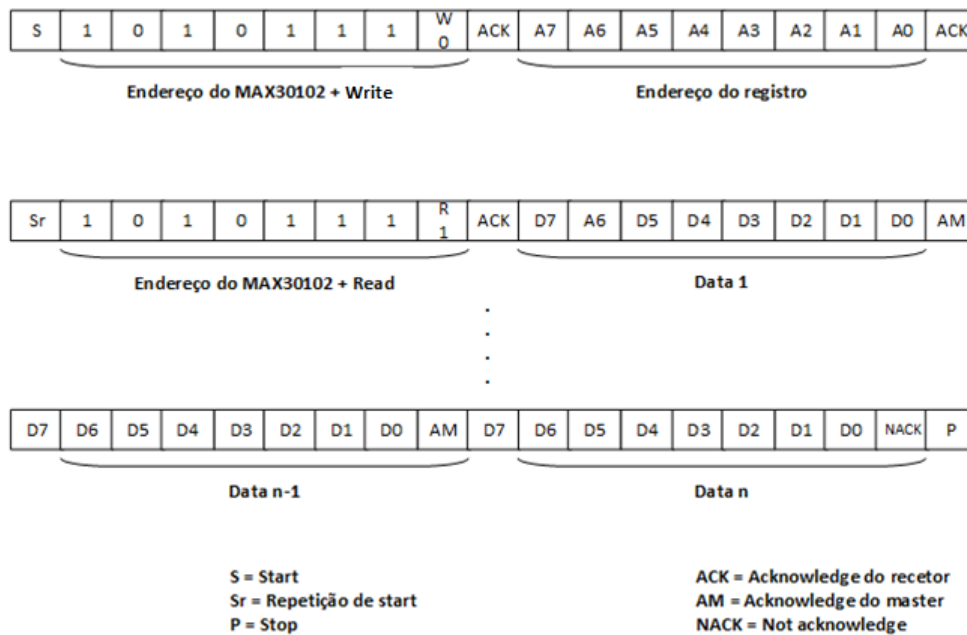


Figura 5.5: Leitura de vários bytes de dados do MAX30102

5.2.2 Sensor de temperatura

O MLX90614 inclui uma interface de comunicação I^2C que permite a comunicação entre um *master* (MCU) e um *slave* (sensor de temperatura). O MCU tem acesso a todos os registos presentes na RAM (*random-access memory*) e na EEPROM (*electrical-erasable programmable read-only memory*) do MLX90614 mas apenas tem permissão para escrever em 9 células da EEPROM, que permitem alterar aspetos específicos deste sensor.

Quando o MCU requisitar uma operação de leitura, o MLX90614 responderá com 16 bits de dados e 8 *code error* bits. Estes últimos 8 bits servem para detetar falhas na receção de dados.

Toda a informação necessária acerca das configurações de fábrica deste elemento pode ser consultada de seguida:

MAX30102 Address 0b1011010x

Leitura LSb = 1

Escrita LSb = 0

Uma operação de transferência de dados para o MLX90614 e uma operação de leitura seguem os esquemas da figura 5.6 e 5.7, respetivamente.

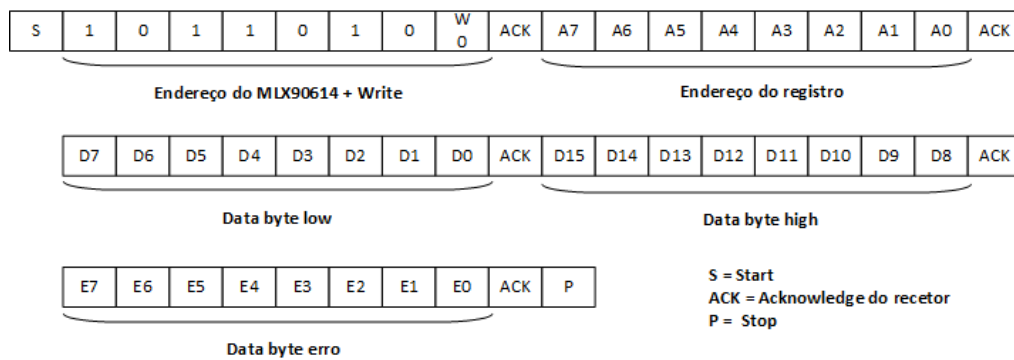


Figura 5.6: Configuração de um registo do MLX90614

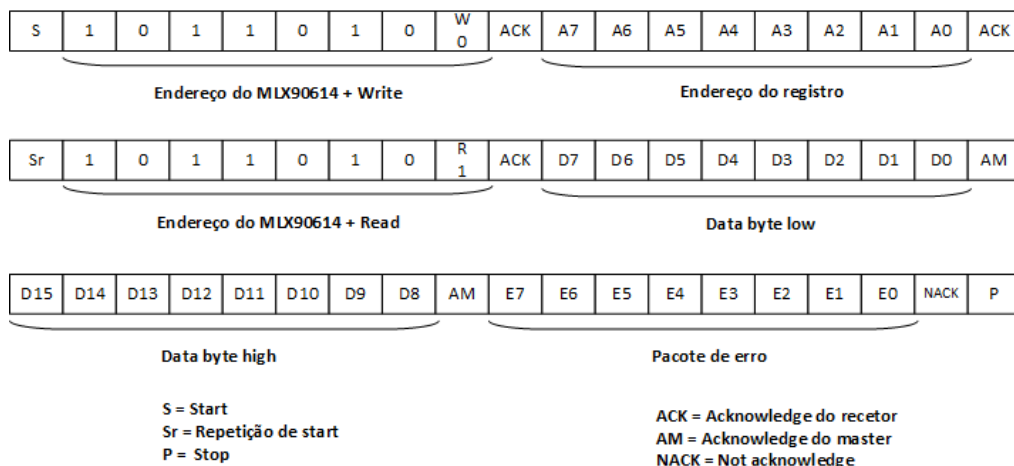


Figura 5.7: Leitura de dados do MLX90614

O *data byte low* reflete a parte fracionária da temperatura em graus Celsius do utilizador, por sua vez, o *data byte high* representa a parte inteira.

5.2.3 Módulo BLE

RS-232 é um protocolo de comunicação ponto a ponto. Isto significa que um dispositivo pode ligar-se apenas a outro mas ambos podem desempenhar a função de transmissor ou recetor. Com este protocolo a informação é enviada sem o auxílio de uma linha com o sinal de relógio. Em vez disso, o recetor e o transmissor estão pré-configurados para funcionarem à mesma velocidade (*baud rate*) e para possuírem os mesmos parâmetros nas tramas de informação. Uma palavra de informação consiste numa sequência de 8 bits, seguida por um bit opcional de paridade e por um ou dois *stop* bits [Vah09].

No desenvolvimento deste projeto foi necessário implementar um *device driver* UART para que o CC1110 pudesse enviar/receber dados do módulo BLE e MatLab, através de uma comunicação RS-232. O *driver code* em questão foi descarregado da Texas Instruments [NF09]. Os parâmetros da interface UART com o MCU CC1110, presentes no ficheiro **options.h**, foram configurados de acordo com a tabela 5.1.

Os parâmetros de comunicação entre o Arduino Leonardo e o módulo BLE mantiveram-se relativamente à versão com o MCU CC1110.

Tabela 5.1: Configurações da UART.

Parâmetro	Valor
UART baudrate	9600
UART parity	None
UART flow control	Off
UART stop bits	1

Cada trama de dados numa comunicação RS-232 começa com um *start* bit. Desta forma, o dispositivo recetor consegue sincronizar o seu relógio para que esteja em fase com o relógio do transmissor, permitindo ao recetor detetar os bits de dados nos intervalos de tempo corretos. A UART do sistema não usou bit de paridade e apenas um *stop* bit, como ilustrado na figura 5.8.



Figura 5.8: Formato da trama RS-232 usada.

A velocidade na transmissão da UART é medida em *bauds*. O *baud-rate* corresponde ao número de símbolos enviados por segundo. Como a comunicação RS-232 codifica um bit, o *baud-rate* é igual ao *bit-rate*. Apesar disso, a taxa de dados líquida é inferior a 9600 bits/seg, uma vez que a trama UART possui um *start* e *stop* bits [Vah09].

5.3 Funcionamento do oxímetro

Antes de adquirir qualquer amostra proveniente do MAX30102 é necessário configurar este módulo para que funcione da forma mais eficiente no contexto deste projeto. Todas as possíveis configurações podem ser detalhadamente analisadas através do *datasheet* fornecido pela Maxim [Max15].

5.3.1 Configurações

O MAX30102 fornece ao técnico a possibilidade de configurar parâmetros relativos à frequência de amostragem, resolução das amostras, FIFO, amplitude do pulso aplicado aos LEDs entre outros.

FIFO

O registo da figura 5.9 permite configurar aspetos relativos ao FIFO circular do oxímetro.

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
FIFO Configuration	SMP_AVE[2:0]			FIFO_ROL LOVER_EN	FIFO_A_FULL[3:0]				0x08	0x00	R/W

Figura 5.9: Registo FIFO.

Este registo possui os seguintes campos:

- **SMP_AVE[2:0]** - possibilita a aplicação de uma média a 1, 2, 4, 8, 16 ou 32 amostras consecutivas extraídas pelo oxímetro. No desenvolvimento deste projeto foi realizada uma média entre cada sequência de 8 amostras;
- **FIFO_ROLLOVER_EN** - permite estabelecer o que sucede se o FIFO ficar cheio de amostras. Se o FIFO encher não serão recolhidas novas amostras até serem obtidas as mais antigas;
- **FIFO_A_FULL[3:0]** - configura o número de amostras (1 a 17 amostras), disponíveis no FIFO, que são necessárias para ser ativada uma interrupção de alerta; 17 amostras (metade da capacidade do FIFO) para este sistema.

Modo de funcionamento

O registo da figura 5.10 permite configurar o oxímetro relativamente ao seu modo de funcionamento.

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
Mode Configuration	SHDN	RESET				MODE[2:0]			0x09	0x00	R/W

Figura 5.10: Registo modo de funcionamento.

Este registo possui os seguintes campos:

- **SHDN** - conduz o oxímetro a entrar num estado de inatividade, onde as funcionalidades internas do oxímetro continuam a ser executadas (excepto as interrupções) num estado de poupança energética. Esta função não foi utilizada pelo sistema;
- **RESET** - permite reiniciar todas as configurações, *thresholds* e esvaziar o FIFO. Esta função não foi utilizada pelo sistema;
- **MODE[2:0]** - estabelece se será ativado o modo de ritmo cardíaco (apenas o LED vermelho é utilizado) ou S_pO_2 (ambos os LEDs são utilizados). Como se pretende a extração tanto do ritmo cardíaco como do S_pO_2 , este campo vai ser colocado no modo S_pO_2 .

Modo S_pO_2

O registo da figura 5.11 permite configurar os parâmetros relacionados com o modo S_pO_2 .

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
S_pO_2 Configuration		SPO2_ADC_RGE[1:0]		SPO2_SR[2:0]		LED_PW[1:0]			0x0A	0x00	R/W

Figura 5.11: Registo S_pO_2 .

Este registo possui os seguintes campos:

- **SPO2_ADC_RGE[1:0]** - estabelece a escala total da ADC (2048, 4096, 8192 ou 16384 nA). De acordo com os testes realizados ao módulo de oximetria, apresentados no capítulo 7, configurou-se a escala total da ADC para 8192 nA;
- **SPO2_SR[2:0]** - impõe a frequência de amostragem do oxímetro (50, 100, 200, 400, 800, 1000, 1600 ou 3200 Hz). O oxímetro foi configurado para recolher amostras com uma frequência de 400 Hz. Apesar disso, como foi realizada uma média entre 8 amostras, a frequência de amostragem real é 50 Hz;
- **LED_PW[1:0]** - estabelece a largura de pulso aplicado a cada LED e consequentemente a resolução das amostras (15, 16, 17 ou 18 bits). De acordo com os testes realizados ao módulo de oximetria, apresentados no capítulo 7, configurou-se a resolução das amostras para 18 bits.

LEDs

Os registos da figura 5.12 permitem configurar os LEDs vermelho e infravermelho.

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
LED Pulse Amplitude	LED1_PA[7:0]								0x0C	0x00	R/W
	LED2_PA[7:0]								0x0D	0x00	R/W

Figura 5.12: Registos do LED vermelho e infravermelho.

Este registo possui os seguintes campos:

- **LED1_PA[7:0]** e **LED2_PA[7:0]** - permitem configurar a amplitude do pulso (entre 0 e 50 mA) aplicada ao LED vermelho e infravermelho, respetivamente. De acordo com os testes de desempenho realizados ao módulo de oximetria, apresentados no capítulo 7, configurou-se estes campos com 15.9 mA.

5.3.2 Aquisição das amostras do oxímetro

Depois de ligado e configurado, o módulo de oximetria está pronto a extrair amostras relativas à quantidade de hemoglobina presente no sangue. Este sensor recolhe as amostras de cada LED consoante a frequência de amostragem definida e armazena-as num FIFO circular, que pode reter até 32 amostras, para posteriormente serem enviadas para o MCU. A figura 5.13 ilustra a organização das amostras no FIFO.

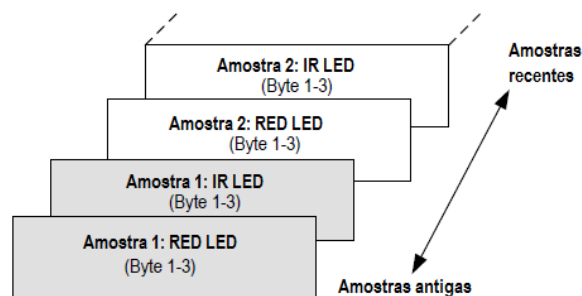


Figura 5.13: Organização dos dados no FIFO do oxímetro

Para saber a quantidade de amostras armazenadas no FIFO é necessária a consulta dos valores fornecidos pelos registos **FIFO_WR_PTR** (*FIFO Write Pointer*) e **FIFO_RD_PTR** (*FIFO Read Pointer*). O primeiro aponta para a localização onde o MAX30102 armazena a próxima amostra no FIFO, avançando uma posição cada vez que é introduzida uma nova amostra, já o segundo refere-se à localização da próxima amostra que tem de ser obtida pelo MCU, avançando uma posição cada vez que uma amostra é lida com sucesso. De seguida é apresentado um pseudo-código, que deve ser implementado em ciclo, relacionado com a leitura das amostras do FIFO.

Primeiro passo - Obter o **FIFO_WR_PTR** e Read **FIFO_RD_PTR**:

```
Start I2C communication;
Read FIFO_WR_PTR;
Read FIFO_RD_PTR;
Stop I2C communication;
```

Segundo passo - Cálculo do número de amostras disponíveis no FIFO:

```
NUM_AVAILABLE_SAMPLES = FIFO_WR_PTR - FIFO_RD_PTR;
NUM_SAMPLES_TO_READ <= NUM_AVAILABLE_SAMPLES ?
```

Terceiro passo - Ler **NUM_SAMPLES_TO_READ** amostras do FIFO:

```
Start I2C communication;
for (i = 0; i < NUM_SAMPLES_TO_READ; i++)
    Read FIFO_RED_DATA (3 bytes);
    Read FIFO_IR_DATA (3 bytes);
Stop I2C communication;
```


5.4 Construção e envio dos pacotes de dados

Após a recolha das amostras ao longo do tempo, o MCU terá de enviar estas amostras, relativas à hemoglobina presente nas artérias e temperatura corporal em pequenos pacotes de dados para o MatLab ou módulo BLE. É importante garantir que o microcontrolador consegue despachar as amostras atempadamente, para que não haja perda de informação na monitorização em tempo real.

5.4.1 Envio de dados para o MatLab

De acordo com o que foi abordado na subsecção 5.3.1, o oxímetro foi configurado para extrair dados com uma frequência de amostragem de 50 Hz e 18 bits por amostra (3 bytes). Isto faz com que cada LED extraia 150 bytes (50×3) de dados por segundo. No entanto, como são utilizados dois LEDs para a extração do ritmo cardíaco e S_pO_2 , o sensor de oximetria fornece um total de 300 bytes de dados por segundo. Como apenas é necessário atualizar a temperatura corporal aproximadamente de segundo em segundo, serão enviados dois bytes (parte inteira e fracionária) por segundo relativos a este sinal vital.

Com os vários bytes de informação que o MCU recebe, inicialmente foram construídos pacotes de dados com a estrutura apresentada pela figura 5.14 e enviados posteriormente através de uma ligação RS-232 para o MatLab no PC.

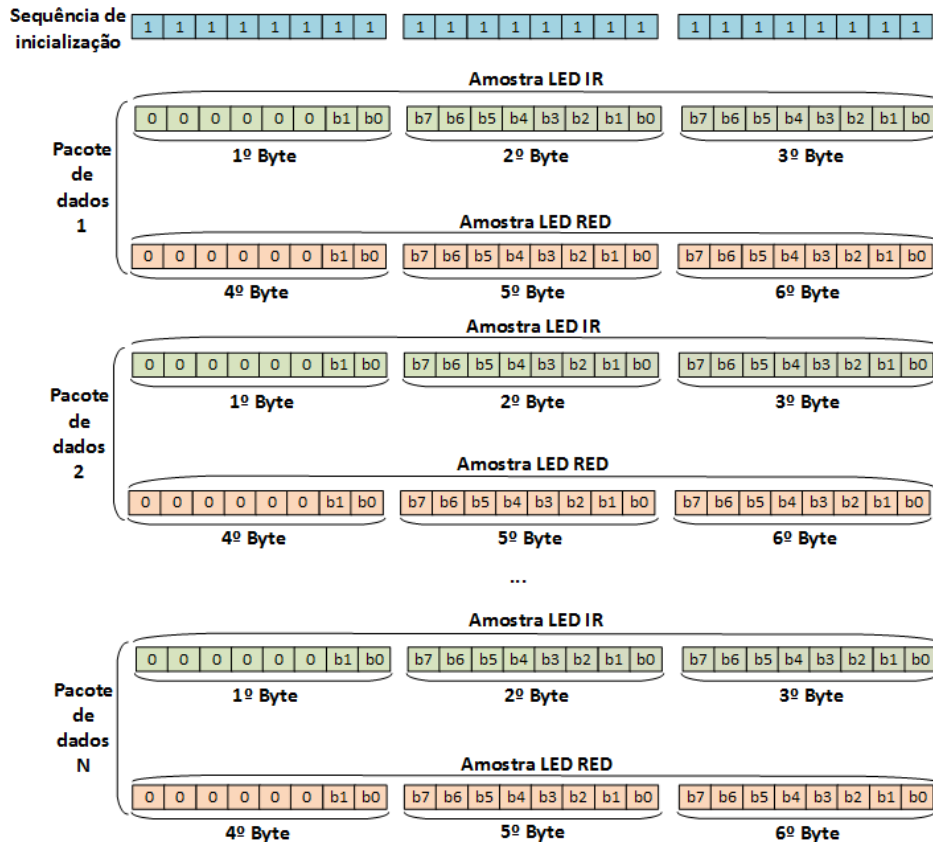


Figura 5.14: Envio dos pacotes de dados relativos à hemoglobina para o MatLab.

5.4.2 Envio de dados para o módulo BLE

Apesar do envio dos pacotes de dados para o MatLab seguir o fomato da figura 5.14 e não apresentar qualquer problema, o mesmo não acontece com ligação ao *smartphone*. O módulo BLE não permite taxas tão elevadas (300 bytes/seg) e portanto os pacotes de dados tiveram de ser otimizados para enviar a mesma informação em menores débitos (75 bytes/seg), como é abordado na secção 6.3. Da mesma forma que no envio dos dados para o MatLab, apenas é necessário atualizar a temperatura corporal aproximadamente de segundo em segundo, portanto, serão enviados dois bytes por segundo relativos a este sinal vital, um byte referente à parte inteira e outro à parte fracionária da temperatura. A influência dos dados relativos à temperatura corporal na taxa de envio de dados para o módulo BLE não é significativa, uma vez que apenas aumenta em 2 bytes/seg o débito solicitado ao módulo BLE.

O envio dos pacotes de dados para o módulo BLE segue o formato apresentado pela figura 5.15

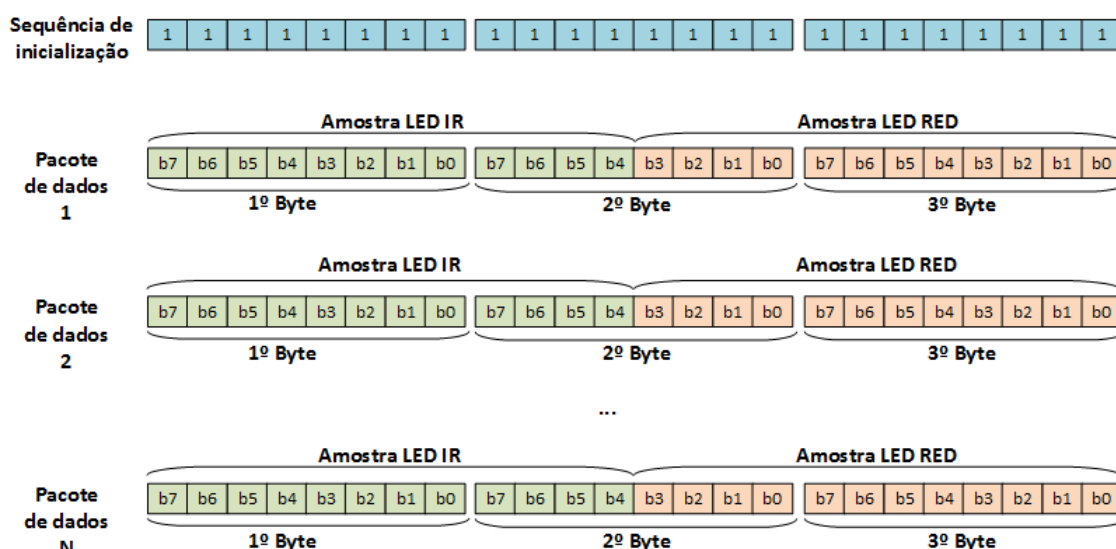


Figura 5.15: Envio de pacotes de dados relativos à hemoglobina para o módulo BLE.

5.5 Sequência de operações do MCU

O fluxograma da figura 5.16 representa a sequência de operações realizada pelos dois microcontroladores utilizados, tanto com ligação ao PC como com ligação à aplicação *Android* através do módulo BLE, uma vez que o processo seguido em ambos é semelhante. Apesar do fluxograma da figura 5.16 apresentar dois blocos de envio dos pacotes de dados, os dois modos de funcionamento não podem estar ativos ao mesmo tempo. Na imagem, colocou-se a ligação com o *smartphone* (através do módulo BLE) a tracejado para indicar que a construção dos pacotes de dados para o MatLab é independente da construção dos pacotes para o módulo BLE. Basicamente existem dois projetos de *software*, um para o envio dos dados para o MatLab e outro para o envio dos dados para o módulo BLE.

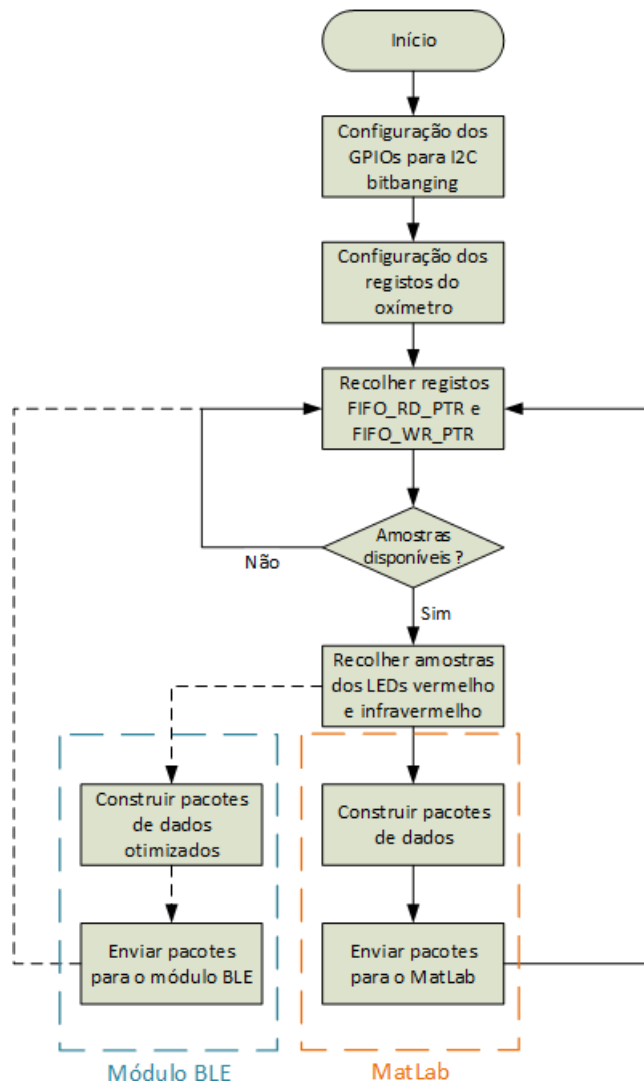


Figura 5.16: Fluxograma das ações do MCU.

5.6 Estrutura de ficheiros

O projeto envolveu o desenvolvimento *software* para dois MCU's diferentes (CC1110 e Arduino) e ainda para duas plataformas (MatLab e *Android*).

5.6.1 Ficheiros do CC1110

O *software* do projeto relativo ao CC1110 para a aquisição das amostras do oxímetro e sensor de temperatura foi desenvolvido através do IAR Embedded Workbench e é composto pelos ficheiros:

- **vital_signs_CC_Ble.c:** Este ficheiro é a Main do *software* do sistema de aquisição de dados com ligação ao *smartphone*. Aqui são realizadas todas as configurações do

oxímetro, leituras das amostras, assim como a posterior construção e envio dos pacotes de informação para o módulo BLE;

- **vital_signs_CC_Mat.c:** Este ficheiro é a Main do *software* do sistema de aquisição de dados com ligação ao MatLab no PC. Aqui são realizadas todas as configurações do oxímetro, leituras das amostras, assim como a posterior construção e envio dos pacotes de informação pela UART;
- **functions.h:** Este ficheiro contém as funções utilizadas pelo `vital_signs_MCU.c`, como a comunicação *I²C bit-bangig* e a construção de pacotes de dados;
- **uart.c:** Este ficheiro suporta o controlador da UART compatível com o CC1110, como a configuração da UART e interrupções de envio/receção de dados;
- **uart.h:** Este ficheiro *header* suporta o controlador da UART, declarando todas as variáveis globais, macros e protótipos das funções usadas pelo ficheiro `uart.c`;
- **uart_intfc.c:** Este ficheiro suporta as funções API (*application programming interface*) usadas na comunicação RS-232, como por exemplo as funções para gestão do FIFO de dados e transmissão de informação;
- **uart_intfc.h:** Este ficheiro *header* suporta as funções API da comunicação RS-232, declarando os protótipos das funções usadas pelo ficheiro `uart_intfc.c`;
- **options.h:** Este ficheiro *header* define as opções do controlador da comunicação RS-232, nomeadamente na indicação do tamanho do *buffer* de transmissão/receção, baudrate, paridade, *stop bits* e *flow control*. Além disso, é especificado o módulo UART do CC1110 usado para a comunicação do projeto.

5.6.2 Ficheiros do Arduino

No desenvolvimento do *software* para o Arduino Leonardo utilizou-se o IDE *open-source* do Arduino e o projeto para aquisição das amostras do oxímetro e sensor de temperatura é composto pelos ficheiros:

- **vital_signs_Ard_Ble.ino:** Este ficheiro agrega todas as bibliotecas necessárias para as comunicações RS-232 e *I²C*. Além disso, é aqui que é desenvolvido todo o código para configuração do oxímetro, receção das amostras e envio das mesmas para o módulo BLE.
- **vital_signs_Ard_Mat.ino:** Este ficheiro agrega todas as bibliotecas necessárias para as comunicações RS-232 e *I²C*. Além disso, é aqui que é desenvolvido todo o código para configuração do oxímetro, receção das amostras e envio das mesmas para o MatLab no PC.

Neste capítulo apresentou-se o sistema de aquisição de dados referentes à hemoglobina nas artérias e temperatura corporal, assim como o seu envio para o módulo BLE. No próximo capítulo realizar-se-á uma descrição sobre o desenvolvimento da App *Android*, nomeadamente o processamento dos dados e como é feita a conexão entre o sistema de aquisição de amostras e o dispositivo móvel.

Capítulo 6

Desenvolvimento da aplicação móvel

O *smartphone Android* foi o dispositivo escolhido para desenvolver esta parte do projeto devido à disponibilidade do seu *kit* de desenvolvimento, documentação, biblioteca de classes e à possibilidade de desenvolver as aplicações em inúmeras plataformas (Linux, Mac OS e Windows). O *smartphone* de teste utilizado é o Wiko Rainbow Lite que possui a versão Android 5.0, no entanto o projeto é compatível com outros dispositivos *Android* com versão 5.0 ou superior.

Uma vez que o objetivo da aplicação móvel é receber dados sobre os sinais vitais de um utilizador via *Bluetooth low energy* e exibí-los no ecrã, deve ser capaz de verificar em primeiro lugar se o dispositivo usado suporta este tipo de comunicação. Se tal for verificado deve ligá-lo para se conectar com o sistema físico e assim, receber os dados para finalmente os exibir de forma perceptível.

6.1 Fundamentos da programação *Android*

As aplicações *Android* são construídas em linguagem Java e compiladas com a ajuda da ferramenta Android SDK. Depois disso é gerado um ficheiro *.apk* que contém toda a informação sobre o projeto e que permite instalar esta aplicação em qualquer dispositivo *Android* [And16].

O sistema operativo *Android* utiliza um sistema Linux multiutilizador. A cada aplicação é atribuído um *user ID* único e apenas este ID pode aceder a dados e permissões da aplicação.

6.1.1 Atividades

Uma atividade permite interações com o utilizador através do ecrã. Este elemento é responsável pela criação de janelas do projeto, que o programador pode alterar da forma que pretender, nomeadamente mudando o fundo, introduzindo caixas de texto, botões, entre outros componentes [Mon14].

Uma aplicação possui geralmente várias atividades e cada atividade pode iniciar outra. Se tal acontecer, a atividade anterior é interrompida mas conservada pelo sistema numa pilha de retorno [Mon14].

O diagrama da figura 6.1 revela o ciclo de vida que uma atividade (representadas pelos blocos coloridos) assim como todos os métodos disponíveis (representados pelos retângulos cinzentos) que o programador pode implementar para realizar qualquer operação que ache conveniente entre atividades.

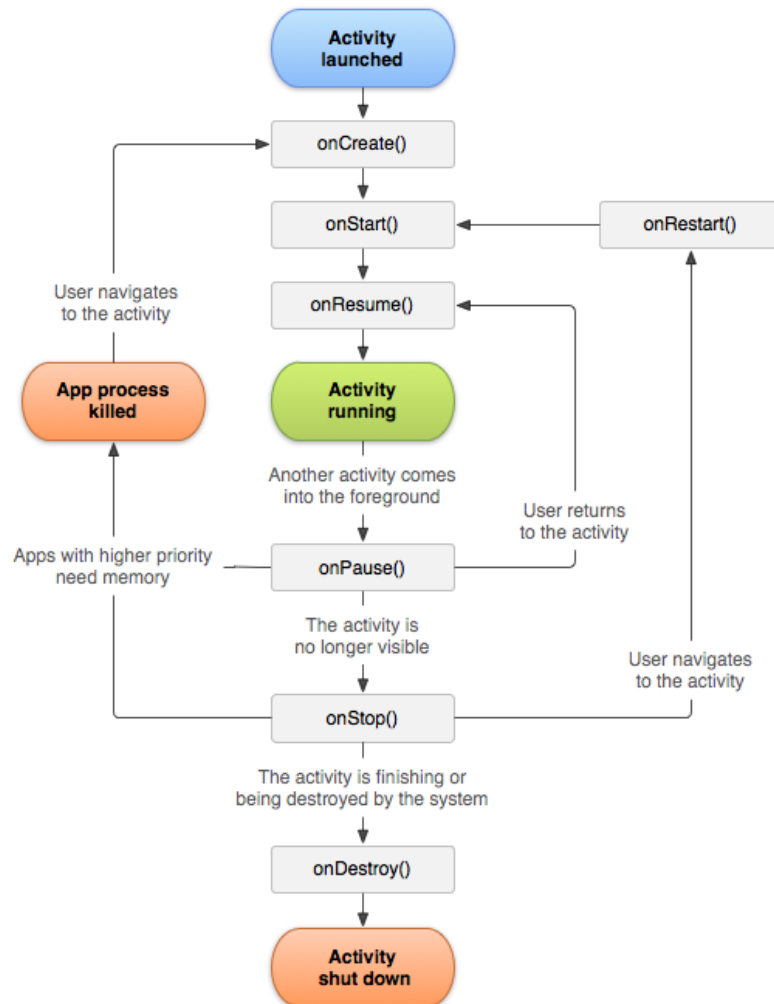


Figura 6.1: Ciclo de vida de uma atividade [Jav16].

6.1.2 Ficheiro XML - *layout*

Um *layout* define a estrutura visual da interface com o utilizador, tal como a UI para uma atividade.

A vantagem de declarar a UI em XML assenta no facto de se conseguir separar a parte visual da aplicação do código que controla o seu comportamento. A descrição da UI é externa ao código da aplicação, possibilitando a sua modificação sem afetar o código nem sendo necessário recompilar o projeto [Mon14].

Cada ficheiro *layout* deve conter exatamente um elemento *root* (objeto *view*), atuando como o bloco construtor básico para os componentes da UI. Depois de se definir este elemento é possível a introdução de *layouts* adicionais ou *widgets* para gradualmente se construir a hierarquia que define toda a camada visual da aplicação móvel [Mon14].

Cada elemento de uma atividade deve ser referenciado por um ID para possibilitar a alteração de alguma das suas características.

Enquanto que a UI pode ser desenvolvida através do ficheiro XML, a aplicação em si é principalmente composta por código Java. Estes ficheiros podem ser localizados nas pastas **src** e **java** do projeto.

6.1.3 *Manifest*

Todas as aplicações *Android* tem de conter um ficheiro chamado de **AndroidManifest** na pasta do projeto que descreve a natureza da aplicação, assim como de cada um dos seus componentes. Este ficheiro possui as informações necessárias sobre o nível mínimo de API da aplicação móvel, as API *library* usadas, as permissões da aplicação, assim como recursos de *software* e *hardware* que possivelmente a aplicação poderá necessitar (*Bluetooth*, câmara, etc) [And16].

6.1.4 *Gradle*

Existe um ficheiro **build.gradle** para cada módulo do projeto referente à App *Android*, assim como um para todo o projeto. É aqui que todas as dependências estão listadas, usadas pelo Android Studio para compilar e fazer *build* à aplicação *Android* [And16].

6.1.5 *GAP*

O GAP (*generic access profile*) controla as conexões assim como o anúncio dos dispositivos *Bluetooth*, ou seja, permite que um dispositivo fique visível para os restantes e determina se dois dispositivos podem interagir um com o outro [IE16].

O GAP define ainda qual dos dois dispositivos será o central (*tablet*, telemóvel, etc) e qual será o periférico [IE16].

6.1.6 *GATT*

O GATT (*generic attribute profile*) define a forma como dois dispositivos BLE trocam informação e é baseado em objetos de alto nível chamados: perfis, serviços e características. Para isso, faz uso do protocolo AAT (*attribute protocol*), usado para armazenar serviços, características e dados numa tabela de ID's. O GATT (figura 6.2) é ativado cada vez que uma conexão é estabelecida entre dois dispositivos e estas ligações são exclusivas, ou seja, cada periférico BLE apenas se pode conectar a um dispositivo central ao mesmo tempo [IE16].

Quando o periférico se conecta ao dispositivo central interrompe imediatamente o seu processo de *broadcast*, o que significa que nenhum outro dispositivo conseguirá vê-lo até a sua conexão ser interrompida.

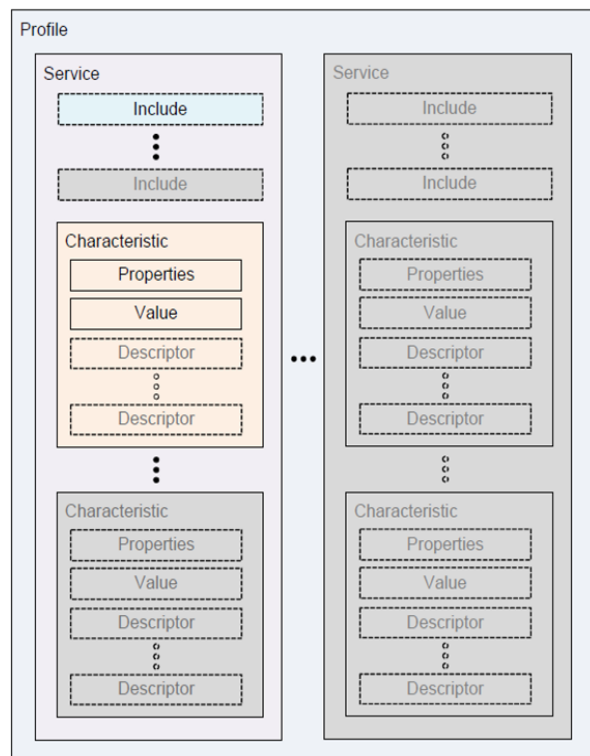


Figura 6.2: Especificações do BLE GATT [Blu16].

Serviços

Os serviços em BLE são usados para dividir os dados em entidades lógicas e características. Um serviço pode ter uma ou mais características e cada serviço distingue-se de outro através de um ID chamado UUID (*universally unique identifier*), composto por 16 ou 128 bits [IE16].

Características

O nível mais baixo do GATT são as características, que contém um único ponteiro para dados. Tal como os serviços, cada característica distingue-se de outra através de um UUID de 16 ou 128 bits [IE16].

Uma vez que é permitido escrever dados numa característica, é possível a troca de dados entre dois dispositivos. Para tal, é necessário implementar um serviço UART personalizado e duas características; uma para o canal TX (*transmitter*) e outra para o canal RX (*receiver*). Um serviço UART foi utilizado pela aplicação *Android* deste projeto [IE16].

6.2 Bluetooth low energy

De seguida é apresentado o procedimento a seguir para incorporar o protocolo BLE num projeto *Android*, assim como o emparelhamento entre dois dispositivos.

6.2.1 Configurações

Uma vez que a aplicação do projeto necessita de uma comunicação BLE, esta permissão tem de ser incluída no ficheiro **AndroidManifest.xml**. Isto é necessário porque é preciso solicitar uma conexão com outro dispositivo, aceitar essa conexão e transferir dados através do *Bluetooth*. Para além disso, é ainda necessário a permissão **BLUETOOTH_ADMIN** se se pretender mudar as definições *Bluetooth*.

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

O uso das funções relativas ao *Bluetooth* requer que as permissões anteriormente abordadas estejam ativas e que o pacote **android.bluetooth** seja incluído no código Java. Depois disto, deve-se verificar se o dispositivo em questão suporta ligação *Bluetooth* através da função **getDefaultAdapter()**. Se não for este o caso, todos os recursos serão desativados e a aplicação não poderá ser usada. Por outro lado se o dispositivo suportar esta comunicação, deve-se verificar de seguida se o *Bluetooth* está ligado, usando a função **isEnabled()**. Por fim, e no caso do *Bluetooth* se encontrar desligado, o utilizador é convidado a ativá-lo sem sair da aplicação com o auxílio da função **startActivityForResult()**.

6.2.2 Ligação com outros dispositivos

Com a inicialização do *Bluetooth* concluída, deve-se começar a procura de dispositivos BLE. Para isso, é necessário que se recorra ao método **startLeScan()** com o parâmetro **BluetoothAdapter.LeScanCallback**.

```
scanCallback = new BluetoothAdapter.LeScanCallback()
{
    //Called when a device is found
    @Override
    public void onLeScan(BluetoothDevice bluetoothDevice, int i, byte[] bytes)
    {
        main.writeLine("Found device: " + bluetoothDevice.getAddress());

        //Check if the device has the UART service.
        if(parseUUIDs(bytes).contains(UART_UUID))
        {
            //Found a device, stop the scan.
            adapter.stopLeScan(scanCallback);
            main.writeLine("Found UART service!");

            //Connect to the device
            gatt = bluetoothDevice.connectGatt(null, true, callback);
        }
    }
};
```

No processo de *scan* de outros dispositivos que se encontram dentro da zona de alcance a aplicação solicita informações sobre cada um deles. Quando um dispositivo é descoberto responde ao dispositivo que realizou o *scan* com o seu nome *Bluetooth*, classe e endereço MAC (*media access control*). Depois do processo de *scan* estar completo, o método **stopLeScan()** é chamado para terminar a procura de dispositivos.

6.3 Receção dos dados

Finalizado o processo de emparelhamento dos dois dispositivos BLE, o *Android* está pronto para começar a receber dados. É necessário ter em atenção as taxas de transferência de dados que o módulo BLE consegue fornecer, uma vez que é necessário enviar dados constantemente. É necessário a receção (sem perdas) de novas amostras relativas à quantidade de luz refletida pela hemoglobina nas artérias para ser possível a extração de valores de ritmo cardíaco e S_pO_2 em tempo real.

Idealmente seria necessário transferir 300 bytes/seg para o sistema executar todas as operações como planeado no capítulo 5, uma vez que é necessário o envio de dados relativos a 2 LEDs (vermelho e infravermelho) que fornecem 50 amostras/seg (configurado pelo utilizador) com 3 bytes de tamanho em cada amostra. Um teste foi realizado e constatou-se que a informação enviada pelo módulo BLE não era totalmente recebida pelo *smartphone*. Por essa razão, decidiu-se desenvolver uma aplicação *Android* que exibisse ao utilizador os débitos de informação recebidos, assim como um simples código no MCU que variava a quantidade de dados enviados por segundo para o módulo BLE. Depois de alguns testes realizados, observou-se que o módulo BLE saturava e perdia pacotes de informação para taxas superiores a 90 bytes/seg.

As soluções encontradas para a resolução do problema relativo aos débitos fornecidos pelo módulo BLE são apresentadas de seguida.

6.3.1 Aumento do MTU

Os baixos débitos do BlueFruit LE, que impedem a transmissão de todos os pacotes de dados, são fruto do próprio modo de funcionamento do BLE, já que neste tipo de comunicação, o dispositivo entra num estado de inatividade entre cada envio de dados. De uma forma mais detalhada, o protocolo BLE envia apenas (por omissão) tramas com o tamanho do MTU (*maximum transmission unit*) igual a 20 bytes. Depois desses 20 bytes serem enviados, o dispositivo entra em estado de inatividade para poupar energia, não existindo qualquer receção ou envio de dados durante esse momento. Se for necessário o envio de 30 bytes, o protocolo separa esta informação numa trama com 20 bytes e outra com 10 bytes, envia a primeira trama, adormece por uma fração de segundo e depois envia os restantes 10 bytes. Numa aplicação em tempo real com necessidade de grandes débitos é fácil compreender porque razão não é aconselhável a utilização de um dispositivo BLE.

Para solucionar este problema resolveu-se aumentar o tempo de atividade do dispositivo BLE. Para tal, seria necessário requisitar ao dispositivo um MTU maior, já que quando se executa uma operação de envio de dados, esta informação é truncada no tamanho atual do MTU. O pedido de um MTU maior, foi inserido na função **onConnectionStateChange**, desta forma é executado no momento do emparelhamento dos dois dispositivos BLE. O código implementado foi o seguinte:

```
//Try to request large MTU
if(gatt.requestMtu(512))
    main.writeLine("Requested large MTU size");
else
    main.writeLine("Failed to request large MTU size");
```

Apesar do código ter sido implementado, o dispositivo *Android* de teste (Wiko Rainbow Lite) e o módulo BLE (Bluefruit LE UART), usados ao longo deste projeto, não permitiram

a alteração do valor do MTU e portanto não foi possível solucionar o problema através desta técnica.

6.3.2 Otimização dos dados

Devido à impossibilidade de aumentar o MTU do BLE decidiu-se formatar os dados de outra maneira, nomeadamente na criação de uma forma de enviar a mesma informação em menor espaço, uma vez que as amostras ADC recolhidas do oxímetro são compostas por uma forte componente DC. Se esta componente fosse removida das amostras antes de ser enviada e posteriormente adicionada pelo *Android* conseguir-se-ia transmitir a mesma informação em menores débitos. A remoção da componente DC permitiu reduzir a taxa de envio de dados para metade (150 bytes/seg), no entanto ainda é acima dos 90 bytes/seg permitidos pelo módulo BLE. A solução passou pela redução da frequência de amostragem para 25 Hz (suficiente para extração dos sinais vitais). Depois do desenvolvimento e implementação deste método, a velocidade com que o MCU transferia pacotes para o módulo BLE fixou-se nos 75 bytes/seg, tornando possível o envio de todas as amostras do oxímetro para o dispositivo *Android*.

Os pacotes recebidos pelo dispositivo *Android* possuem o formato representado pela figura 6.3 e o modo como são enviados pode ser consultado na secção 5.4.

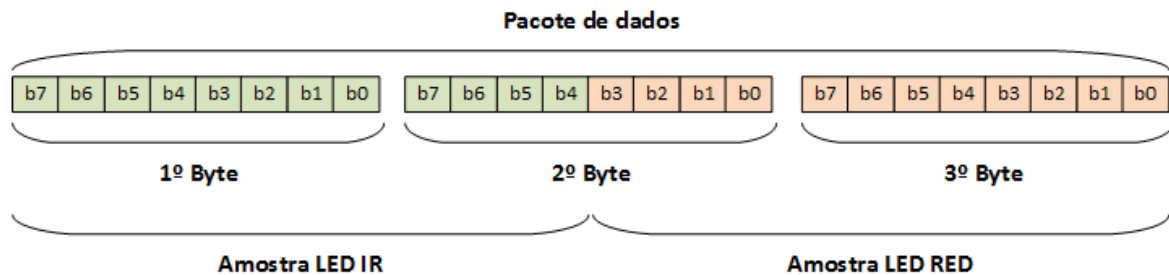


Figura 6.3: Estrutura dos pacotes de dados otimizados.

Tendo conhecimento da estrutura dos pacotes e onde está armazenada a informação, basta aplicar algum tratamento aos bits de cada pacote para retirar as amostras do LED vermelho e infravermelho. No final é necessário adicionar a componente DC que foi retirada anteriormente no MCU.

```
RED_sample = Data_package[11:0];
IR_sample = Data_package[23:12];
RED_sample += RED_DC;
IR_sample += IR_DC;
```

6.4 Visualização dos dados em tempo real

O Android Studio fornece um conjunto de API's de desenho 2D que permitem criar gráficos. O desenho do gráfico referente aos níveis relativos de hemoglobina no sangue foi desenvolvido através da classe CANVAS, que implementa as ações de desenho num mapa de

bits e onde é possível a utilização de vários métodos que tornam mais fácil o controlo de todos os elementos. São necessários 4 componentes para desenhar utilizando o CANVAS:

- **CANVAS** - Indispensável já que possui todos os métodos para desenhar no mapa de bits;
- **Mapa de bits** - Necessário para conter todos pixels;
- **Cor** - Necessária para descrever as cores e estilos do gráfico;
- **Primitivas de desenho**.

Em primeiro lugar, deve configurar-se o CANVAS, nomeadamente criando o bitmap e definindo o seu tamanho, quer em altura quer em comprimento. As seguintes linhas de código transcrevem estas configurações:

```
Bitmap bg = Bitmap.createBitmap(500, 500, Bitmap.Config.ARGB_8888);  
canvas = new Canvas(bg);
```

De seguida deve escolher-se uma cor associada a cada pixel do gráfico, introduzindo as seguintes linhas de código:

```
Paint paint = new Paint();  
paint.setColor(Color.parseColor("#000000"));
```

Por fim, pode construir-se o gráfico através da função **MainVital.canvas.drawLine()**. Esta função une dois pontos, logo, basta percorrer todo o *array* de amostras coletadas para se construir o gráfico do ritmo cardíaco do utilizador. Quando o utilizador analisa o gráfico gerado está a observar a sua atividade cardíaca dos últimos 8 segundos.

6.5 Determinação do ritmo cardíaco e SpO₂

Para além da exibição do sinal do ritmo cardíaco, a aplicação *Android* deve ser capaz de extrair deste sinal os valores instantâneos relativos ao ritmo cardíaco e S_pO_2 do utilizador. O método desenvolvido terá de ser simples e rápido para proporcionar um bom desempenho na monitorização em tempo real. Para se conseguir o menor erro e ruído possível, devem ser aplicados filtros digitais para melhorar significativamente o sinal. Depois da filtragem ser concluída, o processo deve focar-se na extração dos picos superior e inferior associados a cada batimento, para posteriormente se determinar a frequência cardíaca e S_pO_2 do utilizador. Relativamente à temperatura não é necessário qualquer tratamento de dados.

6.5.1 Metodologia

O sinal gerado pelo conjunto de amostras recebidas pelo *smartphone* será similar ao da figura 6.4, onde se podem ver claramente as componentes AC e DC do sinal extraído por um dos LEDs do oxímetro, abordadas no capítulo 2.

Através do sinal da figura 6.4 é possível obter informações sobre o ritmo cardíaco e níveis de S_pO_2 de um utilizador. O ritmo cardíaco pode ser obtido pela frequência deste sinal, ou seja, pela diferença de tempo entre dois picos consecutivos. Para tal, é necessário desenvolver um método que de alguma forma retire os picos do sinal para posteriormente se calcular a diferença de tempo entre eles.

Por outro lado, o S_pO_2 é determinado através da equação 2.3 e portanto, é necessário a análise do sinal proveniente tanto do LED vermelho como do LED infravermelho. De forma similar ao cálculo do ritmo cardíaco, é necessário desenvolver um algoritmo que obtenha o máximo e mínimo de cada sinal.

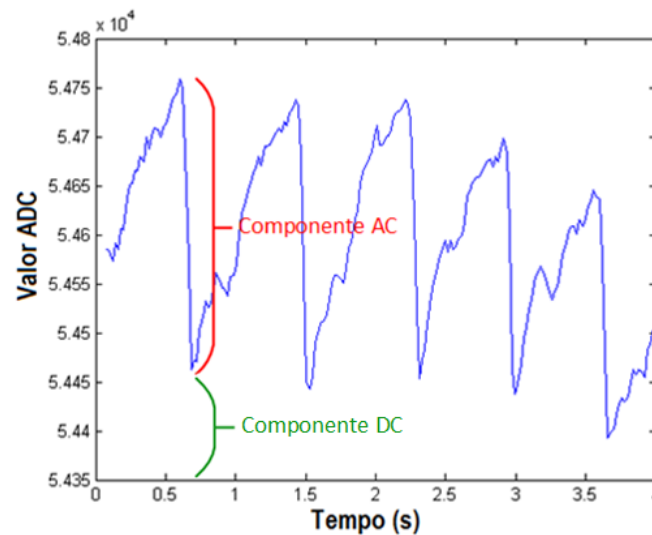


Figura 6.4: Sinal PPG gerado pelas amostras do oxímetro.

A figura 6.5 fornece informação acerca das várias etapas necessárias para retirar os valores finais do ritmo cardíaco e S_pO_2 , a partir do sinal PPG original.



Figura 6.5: Etapas do processamento de sinal.

Implementação dos filtros

Através do MatLab e com o auxílio da ferramenta FDATool, é possível desenvolver de forma tanto rápida como eficaz, filtros para qualquer aplicação. Apesar de projetados no MatLab, os filtros deste projeto foram implementados na aplicação *Android* desenvolvida.

Para obter valores acerca dos níveis de S_pO_2 é necessário achar os valores ADC de cada máximo e mínimo. Foi implementado um filtro digital passa-baixo para se obter um sinal mais suave e conseguir-se achar de forma mais eficiente cada extremo do sinal. Este filtro é um FIR (*finite impulse response*) de ordem 11 do tipo *Window Rectangular*, com cutoff em 0.1 Hz e a sua resposta em frequência é apresentada na figura 6.6.

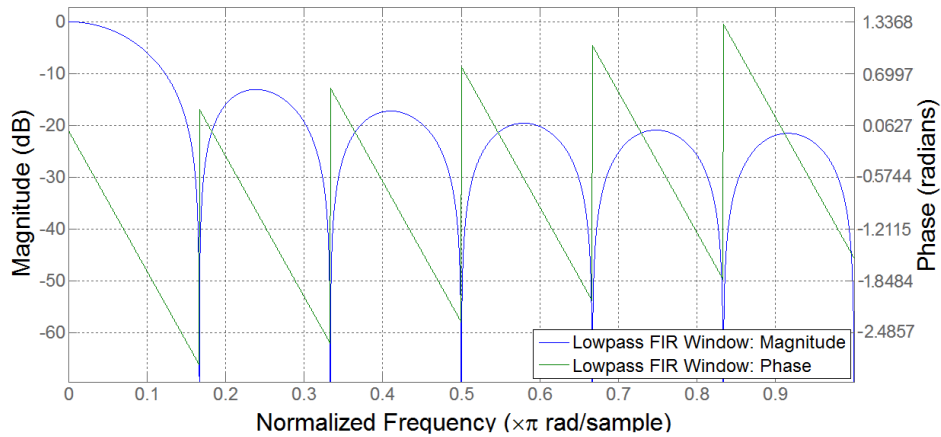


Figura 6.6: Respostas em frequência do filtro passa-baixo.

O sinal original e o sinal filtrado podem ser comparados através da figura 6.7, onde se observa que o sinal depois do filtro passa-baixo mantém as características relevantes do sinal original mas possui um sinal muito mais suave. Desta forma é mais fácil a extração dos valores ADC de cada extremo para posteriormente se proceder ao cálculo do S_pO_2 .

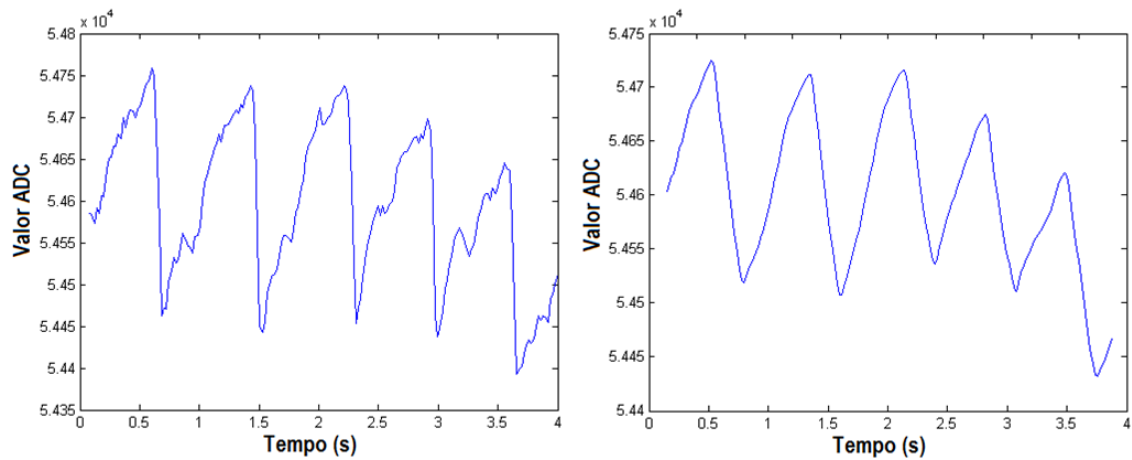


Figura 6.7: Sinal original e sinal após filtro passa-baixo.

Ao contrário do S_pO_2 , para o cálculo do ritmo cardíaco não necessitamos de valores ADC do sinal, em vez disso é necessário saber apenas o intervalo de tempo entre dois batimentos consecutivos. Cada batimento é representado por um pico, portanto, ao aplicar um filtro passa alto ao sinal consegue-se isolar cada um destes extremos, atenuado o restante sinal.

Uma vez mais, com ajuda do FDATool projetou-se um filtro FIR passa-alto de ordem 51 do tipo *Window Rectangular*, com cutoff em 0.1 Hz e a sua resposta em frequência é apresentada na figura 6.8.

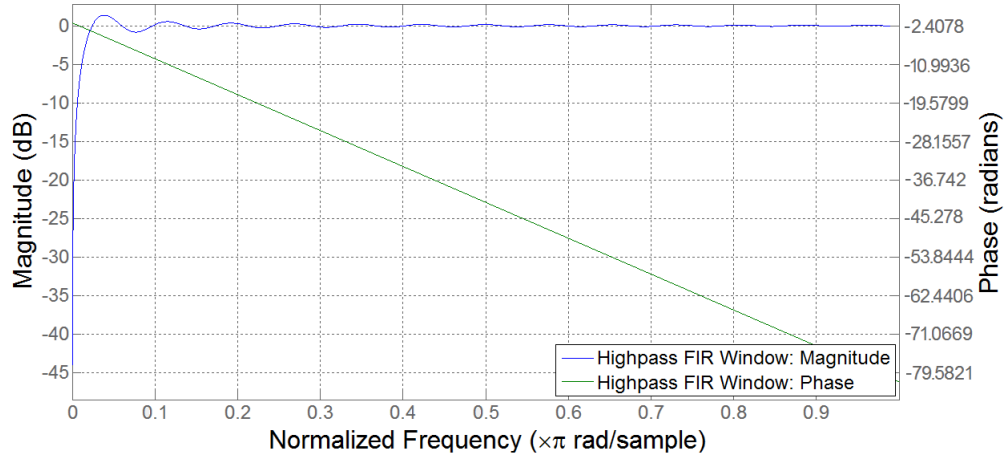


Figura 6.8: Respostas em frequência do filtro passa-alto.

Através da figura 6.9 é possível observar a vantagem de filtrar desta forma o sinal. Os picos referentes a cada batimento ficam muito mais perceptíveis, logo, é mais simples a extração do instante de tempo referente a cada batimento.

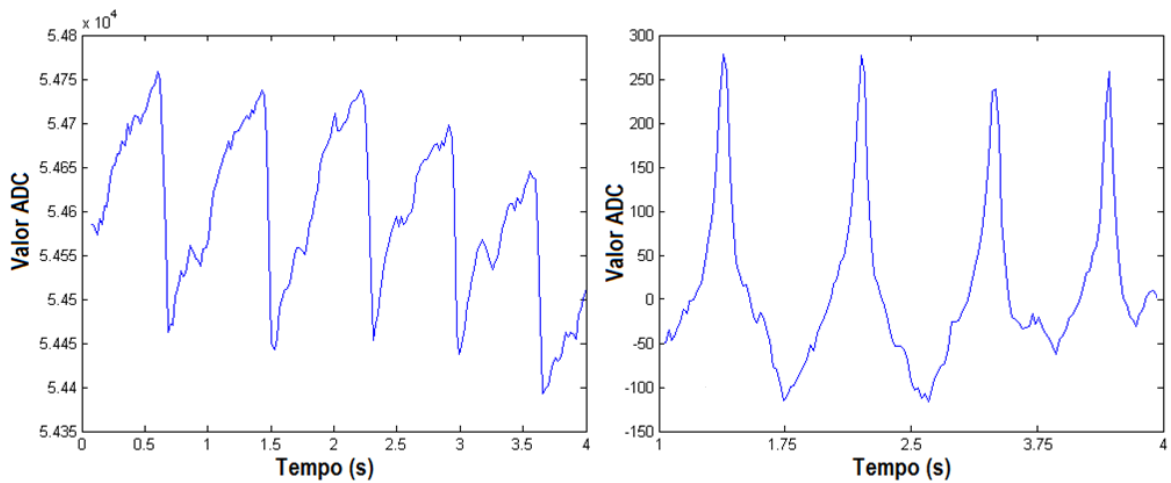


Figura 6.9: Sinal original e sinal após filtro passa-alto.

Calibração dos níveis de SpO2

Com os dados provenientes do oxímetro, referentes às amostras recolhidas pelos LEDs vermelho e infravermelho (valores ADC), é possível obter em cada instante o rácio $R = \frac{AC_R \cdot DC_{IR}}{AC_{IR} \cdot DC_R}$ (equação 2.3), porém, para converter esse rácio em valores de S_pO_2 é necessário recorrer a uma curva de calibração externa. Tendo como base os artigos [MBC94, PM14, HMT⁺15] foi possível achar uma aproximação quadrática desta relação que foi posteriormente linearizada em segmentos. Esta linearização tem o propósito de fornecer uma boa aproximação na conversão dos rácios R em valores S_pO_2 , assim como contribuir para parcelas de cálculo mais simples, uma vez que as expressões provenientes da linearização são de primeira ordem.

Através das figuras 6.10 e 6.11 é possível comparar o modelo quadrático, entre o rácio R e o valor de S_pO_2 , com o modelo linearizado desenvolvido.

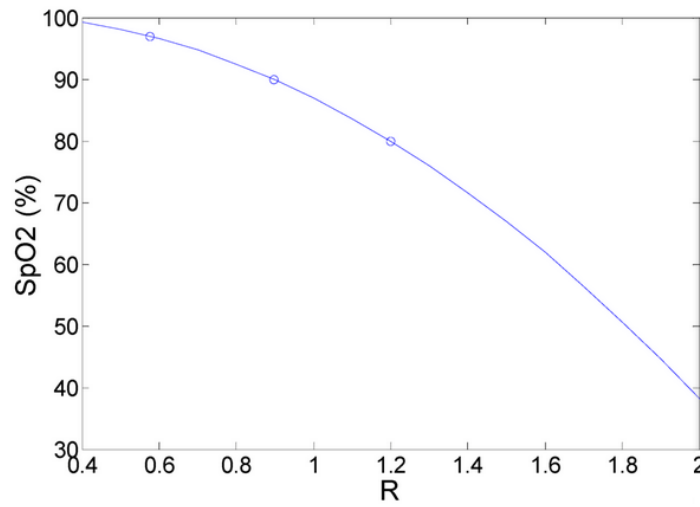


Figura 6.10: Modelo de calibração quadrático [HMT⁺15].

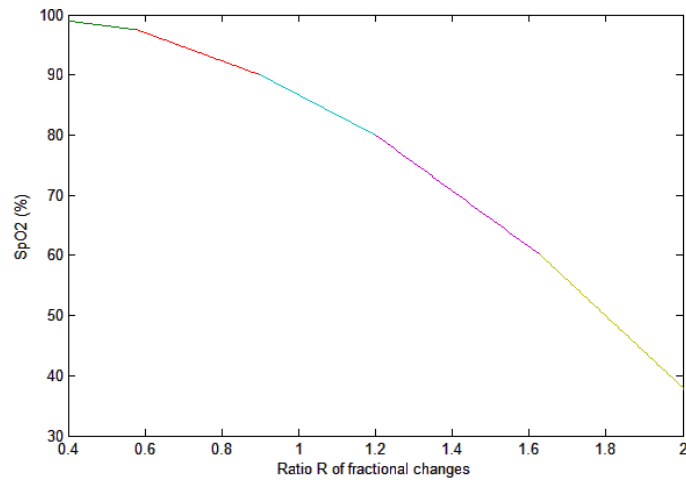


Figura 6.11: Curva de calibração linearizada.

6.6 Sequência de operações da aplicação *Android*

As etapas do algoritmo da aplicação Android relativas à receção, processamento e exibição de informação podem ser consultadas a partir da figura 6.12.

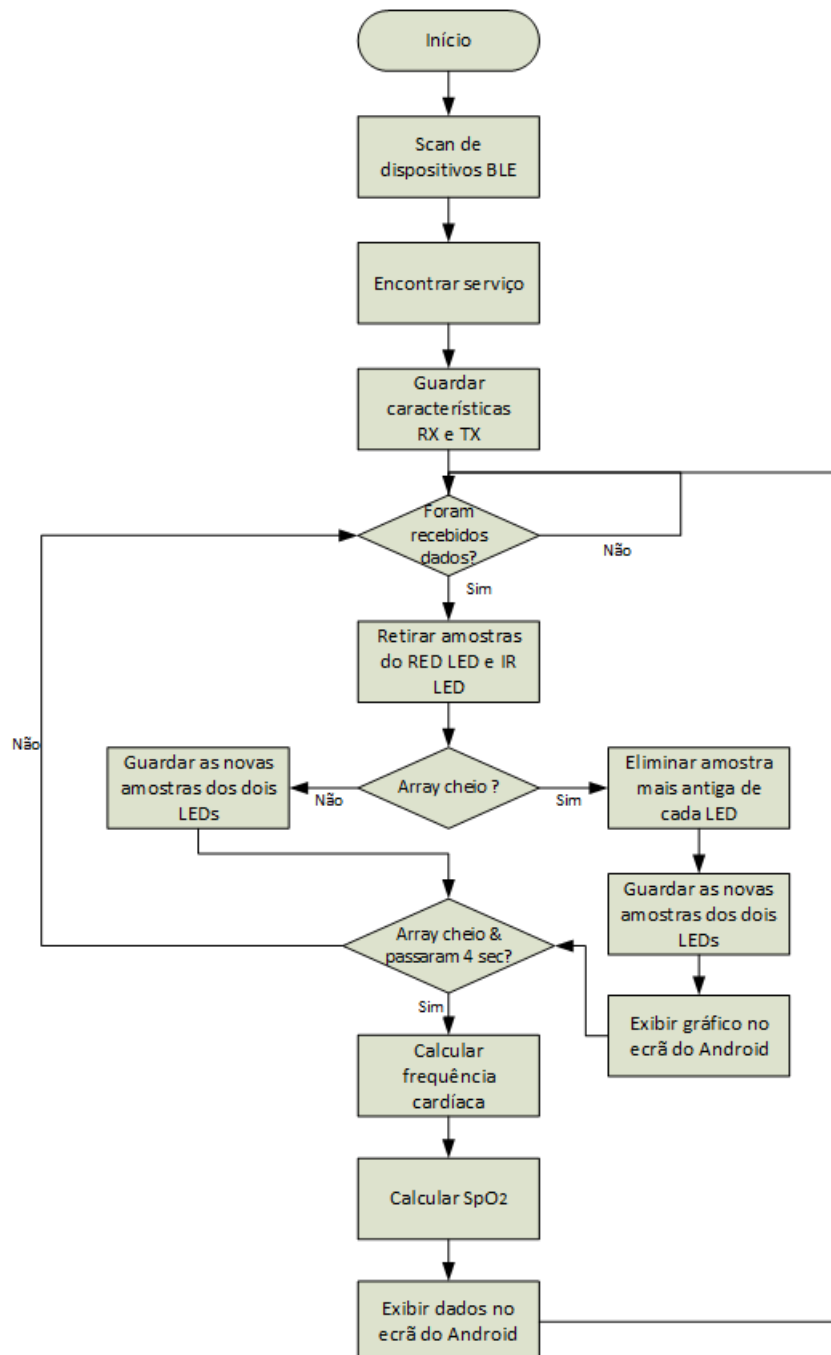


Figura 6.12: Fluxograma das ações da aplicação *Android* desenvolvida.

6.7 Estrutura de ficheiros

O *software* do projeto referente ao desenvolvimento da aplicação *Android* foi produzido através do IDE Android Studio e é composto pelos ficheiros:

- **MainVital.java:** Este ficheiro contém a Main do projeto de *software*, responsável pela inicialização do *Bluetooth* e mudança de estados de execução da aplicação;
- **BluetoothHandler.java:** Este ficheiro suporta todo o protocolo BLE no *smartphone*, nomeadamente o GAP e GATT. Além disso, é aqui que é manipulada toda a informação recebida e onde são aplicados os algoritmos desenvolvidos para a extração dos valores finais de ritmo cardíaco e S_pO_2 ;
- **AndroidManifest.xml:** Este ficheiro é executado antes de qualquer outro e fornece as informações necessárias sobre o sistema *Android*, como as permissões necessárias e o nível mínimo da API *Android* requeridas pela aplicação;
- **activity_data.xml:** Este ficheiro é responsável pela construção e gestão do ecrã onde o gráfico e os valores do ritmo cardíaco, S_pO_2 e temperatura são apresentados;
- **activity_waiting.xml:** Este ficheiro define o *layout* apresentado ao utilizador enquanto que o sistema de aquisição de dados e o *smartphone* estão a estabelecer a comunicação BLE entre eles;
- **build.gradle:** Este ficheiro automatiza e faz a gestão do processo *build* da aplicação.

Toda a informação acerca do desenvolvimento da aplicação *Android* foi descrito neste capítulo. No próximo será apresentado o funcionamento global do sistema e todos os testes de desempenho realizados ao sistema desenvolvido.

Capítulo 7

Resultados e discussão

Neste capítulo são apresentados, numa primeira fase, os testes realizados ao módulo de oximetria utilizado com diferentes conjuntos de parâmetros, para avaliar o seu desempenho com cada alteração. Depois de concluídos, estes testes auxiliaram no processo de escolha do modo de funcionamento do oxímetro. O capítulo inclui também uma explicação detalhada acerca do modo de funcionamento do sistema e uma bateria de testes realizados a um conjunto de indivíduos, utilizando o sistema desenvolvido e dois oxímetros de pulso comerciais. Por fim, é apresentada uma discussão crítica sobre os resultados obtidos.

7.1 Testes de configuração do oxímetro

Numa fase inicial do projeto foi necessário testar o módulo de oximetria para perceber como poderia funcionar. Estes testes foram realizados variando os valores da escala da ADC, resolução das amostras e a amplitude do pulso aplicado a cada LED. Para tal, foi criado um *script* no MatLab que recebia a informação e exibia num gráfico o conjunto das amostras, tornando perceptíveis as alterações em cada parâmetro.

7.1.1 Configuração do modo SpO2

O módulo de oximetria utilizado possibilita a configuração da escala da ADC através do campo SPO2_ADC_RGE do registo **SpO2 configuration** de acordo com a tabela 7.1.

Tabela 7.1: Configurações da escala ADC.

SPO2_ADC_RGE	LSb (pA)	Escala ADC (nA)
00	7.81	2048
01	15.63	4096
10	31.25	8192
11	62.5	16384

A configuração da resolução das amostras, estabelecida pela variação da largura do pulso aplicado a cada LED, é realizada através do campo LED_PW do registo **SpO2 configuration** de acordo com tabela 7.2.

Tabela 7.2: Configurações da resolução das amostras.

LED_PW	Largura do pulso (μ s)	Resolução ADC (bits)
00	69	15
01	118	16
10	215	17
11	411	18

Os resultados relativos à variação dos parâmetros da escala ADC e resolução das amostras podem ser consultados através das representações gráficas abaixo apresentadas.

Começou por se colocar a escala da ADC no seu valor máximo (16384 nA) e a resolução de cada amostra no seu valor mínimo (15 bits). Estes parâmetros foram configurados no módulo de oximetria do sistema resultando no sinal apresentado pela figura 7.1, onde se nota a existência de variações de sangue nas artérias do utilizador mas não é perceptível o sinal cardíaco. Para tentar melhorar o sinal foi mantida a escala da ADC e aumentada a resolução das amostras.

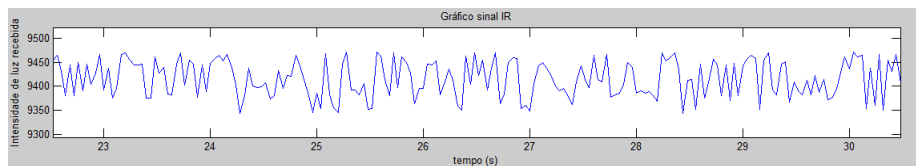


Figura 7.1: Escala ADC 16384 nA e 15 bits de resolução.

Ao observar a figura 7.2, com amostras de 16 bits, já se consegue distinguir a forma do sinal cardíaco, porém, não é regular e por vezes apresenta variações bruscas.

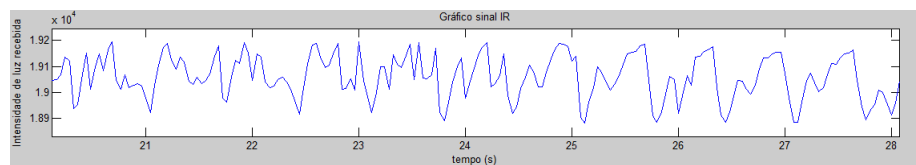


Figura 7.2: Escala ADC 16384 nA e 16 bits de resolução.

Aumentando uma vez mais a resolução das amostras, para 17 bits (figura 7.3), consegue distinguir-se ligeiramente o pulso tal como na figura 7.2, porém, ainda é possível aumentar uma vez mais a resolução (18 bits) para tentar obter um sinal melhor.

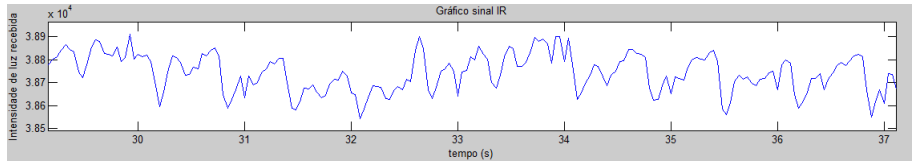


Figura 7.3: Escala ADC 16384 nA e 17 bits de resolução.

Finalmente é possível distinguir cada batimento do coração, observando a figura 7.4. Este sinal aproxima-se do padrão esperado, permitindo criar algoritmos para extração dos valores referentes ao ritmo cardíaco e S_pO_2 .

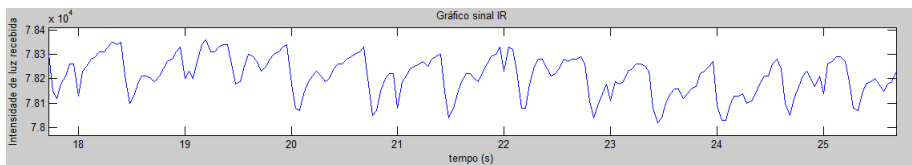


Figura 7.4: Escala ADC 16384 nA e 18 bits de resolução.

Apesar de já se ter encontrado parâmetros para uma boa recolha de amostras relativas à hemoglobina, deve tentar-se melhorar os resultados diminuindo o valor da escala da ADC para 8192 nA e variando de novo o valor da resolução de cada amostra.

As amostras que compõem o sinal da figura 7.5 têm uma resolução de 15 bits (resolução mais baixa). Com esta configuração deixa-se de conseguir distinguir o pulso cardíaco, por isso aumentou-se a resolução para 16 bits.

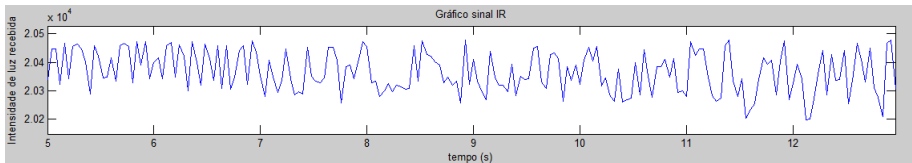


Figura 7.5: Escala ADC 8192 nA e 15 bits de resolução.

Assim como na figura 7.5, não se consegue distinguir o sinal cardíaco da figura 7.6. O uso das resoluções mais baixas nas amostras têm um grande efeito no sinal cardíaco. Isto acontece uma vez que a largura do pulso aplicado a cada LED, ou seja o tempo que está ligado, é menor, resultando num valor ADC menos próximo da realidade.

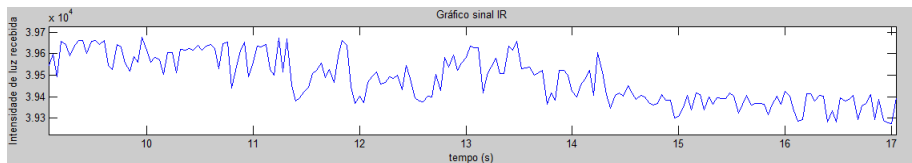


Figura 7.6: Escala ADC 8192 nA e 16 bits de resolução.

Na da figura 7.7, onde as amostras possuem 17 bits de resolução, já se consegue distinguir o pulso cardíaco como acontecia na figura 7.4.

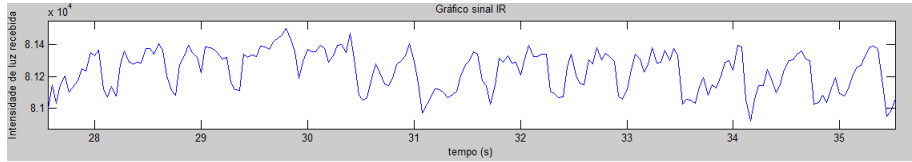


Figura 7.7: Escala ADC 8192 nA e 17 bits de resolução.

Ao colocar a resolução das amostras em 18 bits (valor máximo) verificam-se melhorias em relação às figuras até agora apresentadas. Em comparação com as configurações anteriores, o sinal da figura 7.8 é mais próximo do padrão cardíaco esperado.

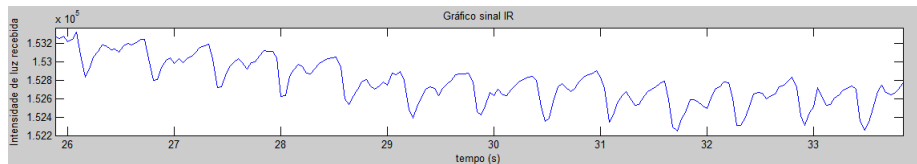


Figura 7.8: Escala ADC 8192 nA e 18 bits de resolução.

Apesar da boa forma de sinal da figura 7.8 realizaram-se testes para os dois valores mais baixos da escala ADC, resultando numa ligeira perda de qualidade do sinal. De seguida são apresentados os resultados para os dois valores da escala ADC restantes, com 18 bits de resolução nas amostras (figuras 7.9 e 7.10). Para resoluções mais baixas o batimento cardíaco era inexistente, tal como nas figuras 7.1 e 7.5 e por essa razão não serão apresentados.

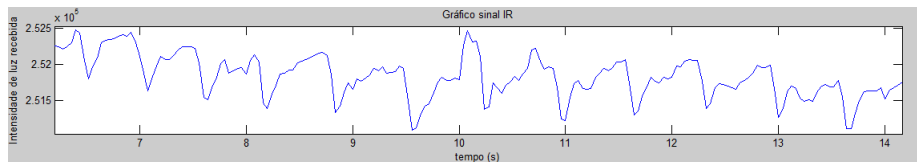


Figura 7.9: Escala ADC 4096 nA e 18 bits de resolução.

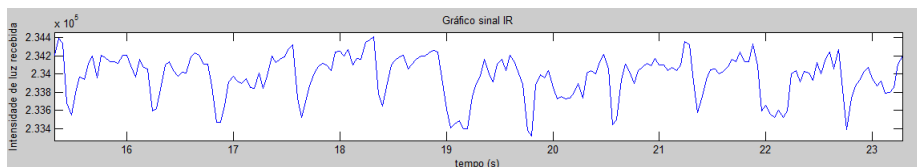


Figura 7.10: Escala ADC 2048 nA e 18 bits de resolução.

Analisando os testes realizados é possível concluir que para a frequência de amostragem usada (50 Hz) é aconselhado que se utilizem as configurações da figura 7.8, uma vez que

apresenta a melhor forma de sinal de todos os testes. Além disso este sinal não revela variações bruscas, permitindo uma correta extração dos valores finais associados ao ritmo cardíaco e S_pO_2 .

7.1.2 Configuração dos LEDs

Além de ser possível configurar o valor da escala ADC e resolução das amostras, é possível indicar a amplitude do pulso aplicado a cada LED. Vários testes foram realizados com valores de corrente fornecidos entre 6 e 20 mA, de modo a determinar a configuração que assegura um melhor desempenho do sistema. Nestes testes, apresentados nas figuras 7.11, 7.12, 7.13, 7.14, 7.15, foram mantidas as configurações da escala ADC e resolução das amostras da figura 7.8.

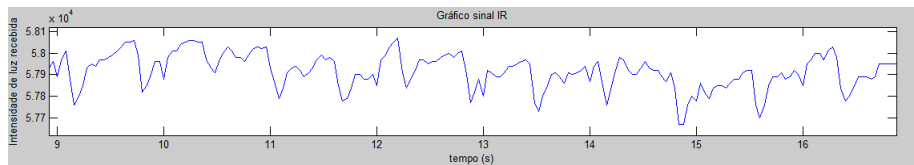


Figura 7.11: Correntes dos LEDs - 6.4 mA.

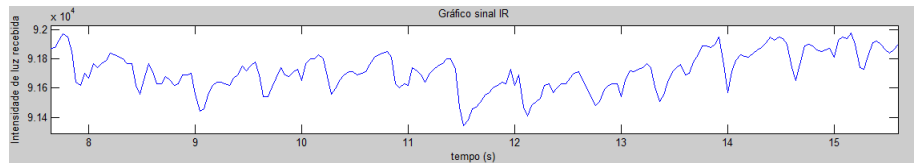


Figura 7.12: Correntes dos LEDs - 9.8 mA.

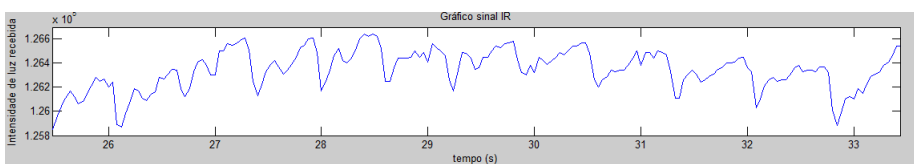


Figura 7.13: Correntes dos LEDs - 12.5 mA.

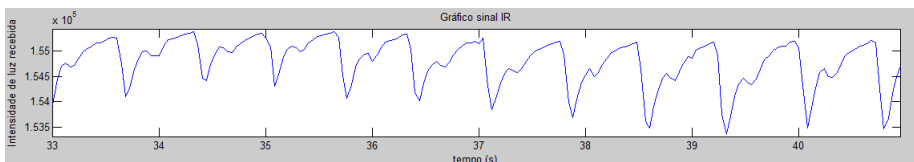


Figura 7.14: Correntes dos LEDs - 15.9 mA.

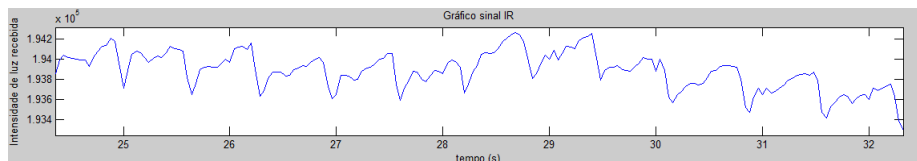


Figura 7.15: Correntes dos LEDs - 19.3 mA.

Através de todos os sinais apresentados é possível detetar com relativa facilidade o sinal cardíaco, no entanto, é visível a qualidade superior das figuras 7.14 e 7.15. Porém, como se pretendeu o menor consumo possível optou-se pela configuração da figura 7.14, portanto, os LEDs vermelho e infravermelho do oxímetro foram alimentados com 15.9 mA.

7.2 Funcionamento do sistema

O primeiro passo para colocar o sistema em funcionamento deve ser a alimentação da placa SmartRF04EB ou do Arduino Leonardo, uma vez que foram realizadas duas provas de conceito com estes dois MCU's. Isto pode ser feito através de uma bateria externa de 9V, um DC jack (4 - 10V) ou por USB, na placa SmartRF04EB. No Arduino Leonardo, a alimentação é realizada através de um DC jack ou de uma ligação micro USB. A figura 7.16 apresenta os tipos de alimentação nas duas placas.



Figura 7.16: Alimentações da SmartRF04EB (esquerda) e do Arduino (direita).

No desenvolvimento do projeto com a placa SmartRF04EB utilizou-se uma alimentação por USB e por micro USB no desenvolvimento com o Arduino Leonardo.

Depois das alimentações estarem asseguradas deve ligar-se a aplicação *Android*, assim como a placa SmartRF04EB ou o Arduino Leonardo, como é ilustrado através da figura 7.17.

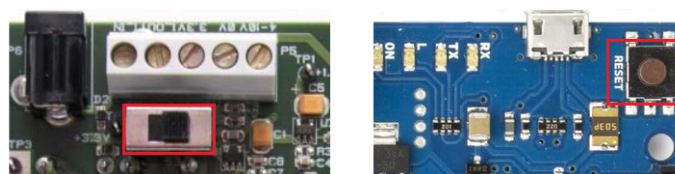


Figura 7.17: Interruptor da SmartRF04EB (esquerda) e *reset* do Arduino (direita).

Com o sistema inicializado, o CC1110/Arduino começará a configurar o oxímetro, recolher

amostras e enviar os pacotes de dados para o módulo BLE. Ao mesmo tempo, este módulo faz *broadcast* das suas informações e o *smartphone* realiza uma procura de dispositivos BLE (figura 7.18) para tentarem estabelecer uma conexão. Depois de estabelecida esta ligação, o módulo BLE começa a converter os pacotes que recebe no protocolo *Bluetooth low energy*, enviando-os posteriormente para o *smartphone*.

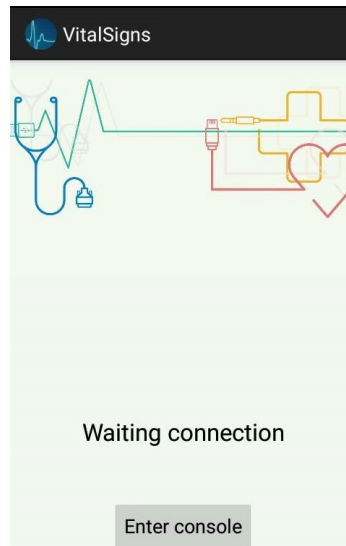


Figura 7.18: Tela de *scan* de dispositivos BLE no *smartphone*.

Do lado do *smartphone*, os pacotes de dados são recebidos, armazenados e processados. Este dispositivo começará a exibir o sinal cardíaco em tempo real e atualizará de 4 em 4 segundos os valores do ritmo cardíaco, S_pO_2 e temperatura corporal no ecrã (figura 7.19).

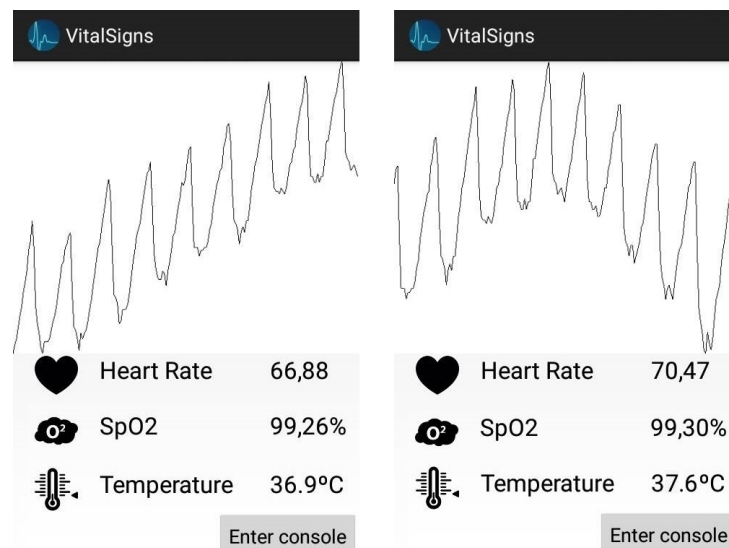


Figura 7.19: Telas de exibição de resultados.

O protótipo laboratorial do sistema é exibido através da figura 7.20, onde são apresentadas tanto a versão com o CC1110 como com o Arduino Leonardo. O sensor de temperatura apenas foi colocado na versão final com o Arduino Leonardo, uma vez que é a configuração principal do projeto.

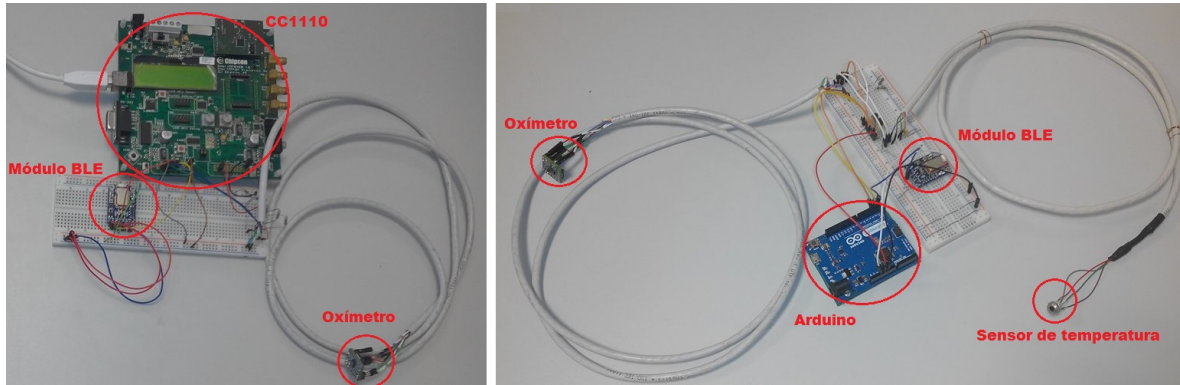


Figura 7.20: Protótipo laboratorial com CC1110 (esquerda) e com Arduino (direita).

Para além da conectividade com um *smartphone Android*, foi desenvolvido em primeiro lugar o sistema de recolha de sinais vitais com ligação ao MatLab do PC (Anexo C). Isto foi realizado inicialmente como forma de validar o funcionamento de todo o sistema de aquisição de dados relativos ao ritmo cardíaco e S_pO_2 .

7.3 Resultados experimentais

Foram realizados vários ensaios experimentais, apresentados de seguida, de modo a testar o sistema desenvolvido em termos de robustez e precisão.

7.3.1 Medição no dedo

Devido à facilidade no manuseamento e à qualidade das amostras recolhidas, o dedo tornou-se no ponto de extração de amostras preferencial durante o desenvolvimento deste projeto.

A utilização do sistema, usando a ponta do dedo, produz um sinal muito similar à da figura 7.21. Com as amostras recolhidas através desta forma é possível retirar com maior facilidade e com maior grau de fiabilidade, valores acerca do ritmo cardíaco e S_pO_2 .

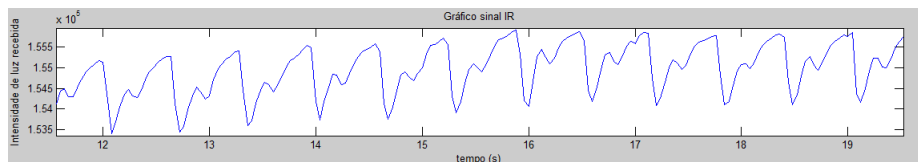


Figura 7.21: Sinal recolhido através do dedo.

Durante a extração das amostras é necessário que o utilizador fixe o seu dedo à sonda de oximetria, uma vez que pequenas variações na posição do dedo em relação à sonda podem provocar distorções no sinal PPG gerado pelo sistema e consequentemente valores de ritmo cardíaco e S_pO_2 errados.

Foram realizados testes de desempenho ao sistema desenvolvido com o auxílio de dois oxímetros de pulso utilizados em ambiente hospitalar. Os oxímetros em questão são o **GIMA OXY-6** e o **CA-MI O2-easy**, com erros idênticos de $\pm 2\%$ tanto na extração da frequência cardíaca como do S_pO_2 [GIM15, CM11]. Estes testes foram realizados a um grupo misto de 35 indivíduos com os três sistemas a operar ao mesmo tempo e foram retirados 4 valores de ritmo cardíaco e S_pO_2 para cada um dos sistemas. Posteriormente, foi realizada uma média das 4 medidas para cada indivíduo, possibilitando um estudo estatístico acerca do desempenho do sistema desenvolvido em comparação com os dois oxímetros de pulso comerciais. A figura 7.22 ilustra um processo de extração de amostras com o sistema desenvolvido e o **GIMA OXY-6**, num dos indivíduos.



Figura 7.22: Extração de amostras com o sistema desenvolvido e o GIMA OXY-6.

Tratamento estatístico

Os resultados dos testes, realizados ao dedo de cada indivíduo do grupo de amostragem com os dois oxímetros comerciais e o sistema desenvolvido, podem ser consultados através da tabela 7.3, assim como o seu erro associado. Durante a realização dos testes, considerou-se como oxímetro de teste 1 (oxímetro comercial 1) o **GIMA OXY-6** e como oxímetro de teste 2 (oxímetro comercial 2) o **CA-MI O2-easy**.

Realizando a média de todos os erros associados às medições de cada indivíduo, concluiu-se que o sistema desenvolvido apresenta um erro (médio) relativo ao ritmo cardíaco de $\pm 2.65\%$ e um erro no S_pO_2 de $\pm 0.28\%$, comparativamente com o **GIMA OXY-6**. Relativamente ao **CA-MI O2-easy**, o sistema possui um erro na extração do ritmo cardíaco de $\pm 1.64\%$ e um erro no S_pO_2 de $\pm 0.29\%$.

Com os dados extraídos a cada indivíduo, foi realizado um estudo estatístico acerca do desempenho do sistema desenvolvido em termos de correlações entre os vários sistemas usados, percentis (25, 50 e 75) e valores médios. Este tratamento estatístico é apresentado nas figuras 7.23, 7.24, 7.25, 7.26 e nas tabelas 7.4, 7.5, 7.6 e 7.7.

Tabela 7.3: Resultados das medições.

Oxímetro do projeto		Oxímetro de teste1		Oxímetro de teste2		Erro oxí. de teste1		Erro oxí. de teste2	
HR (BPM)	SpO2 (%)	HR (BPM)	SpO2 (%)	HR (BPM)	SpO2 (%)	HR (%)	SpO2 (%)	HR (%)	SpO2 (%)
47.75	98.75	46.75	99.00	47.50	99.00	2.1	0.3	0.5	0.3
52.75	98.50	54.00	99.00	55.00	99.00	2.3	0.5	4.1	0.5
55.50	99.00	53.50	99.00	54.25	99.00	3.7	0.0	2.3	0.0
58.25	98.25	55.75	99.00	57.00	98.00	4.5	0.8	2.2	0.3
58.75	98.50	57.25	99.00	58.00	99.00	2.6	0.5	1.3	0.5
59.50	97.75	57.75	99.00	58.75	98.50	3.0	1.3	1.3	0.8
59.75	99.00	58.00	98.75	59.00	99.00	3.0	0.3	1.3	0.0
62.25	99.00	60.50	99.00	60.25	98.75	2.9	0.0	3.3	0.3
63.25	99.00	61.75	99.00	62.75	99.00	2.4	0.0	0.8	0.0
65.00	96.25	62.75	97.00	63.25	95.75	3.6	0.8	2.8	0.5
66.75	98.50	64.00	98.75	64.50	98.25	4.3	0.3	3.5	0.3
67.00	98.75	65.50	98.75	66.00	99.00	2.3	0.0	1.5	0.3
67.25	98.50	65.00	98.75	66.00	98.75	3.5	0.3	1.9	0.3
67.25	98.75	65.25	98.75	65.50	98.50	3.1	0.0	2.7	0.3
67.25	99.00	65.00	98.75	65.25	98.75	3.5	0.3	3.1	0.3
67.75	99.00	66.00	99.00	66.75	99.00	2.7	0.0	1.5	0.0
68.75	98.50	66.00	99.00	69.00	98.00	4.2	0.5	0.4	0.5
70.25	99.00	67.75	99.00	68.00	99.00	3.7	0.0	3.3	0.0
71.00	99.00	70.75	98.50	71.50	98.50	0.4	0.5	0.7	0.5
75.25	97.00	74.00	97.50	73.25	97.50	1.7	0.5	2.7	0.5
75.25	98.75	74.75	98.25	75.75	98.25	0.7	0.5	0.7	0.5
75.75	98.50	75.00	98.75	76.00	99.00	1.0	0.3	0.3	0.5
75.75	98.75	73.25	99.00	74.50	99.00	3.4	0.3	1.7	0.3
77.25	98.75	75.25	98.75	76.25	98.50	2.7	0.0	1.3	0.3
77.50	98.25	75.25	98.25	76.25	98.75	3.0	0.0	1.6	0.5
77.50	98.75	74.25	98.25	75.00	98.00	4.4	0.5	3.3	0.8
78.00	98.75	76.50	99.00	77.50	98.50	2.0	0.3	0.6	0.3
81.75	98.75	80.75	98.75	81.75	98.25	1.2	0.0	0.0	0.5
84.50	98.75	81.50	99.00	82.25	98.75	3.7	0.3	2.7	0.0
86.00	98.25	85.00	98.00	86.00	98.50	1.2	0.3	0.0	0.3
86.25	99.00	85.50	99.00	86.50	99.00	0.9	0.0	0.3	0.0
88.00	98.75	85.75	99.00	96.50	99.00	2.6	0.3	1.7	0.3
93.25	98.00	94.25	98.75	95.25	98.25	1.1	0.8	2.1	0.3
99.25	99.00	96.75	99.00	97.50	99.00	2.6	0.0	1.8	0.0
107.75	99.00	104.50	99.00	106.00	99.00	3.1	0.0	1.7	0.0

Através da figura 7.23 é possível observar a gama de valores das amostras de ritmo cardíaco, extraídas ao grupo de amostragem com o sistema desenvolvido, oxímetro de teste 1 e oxímetro de teste 2.

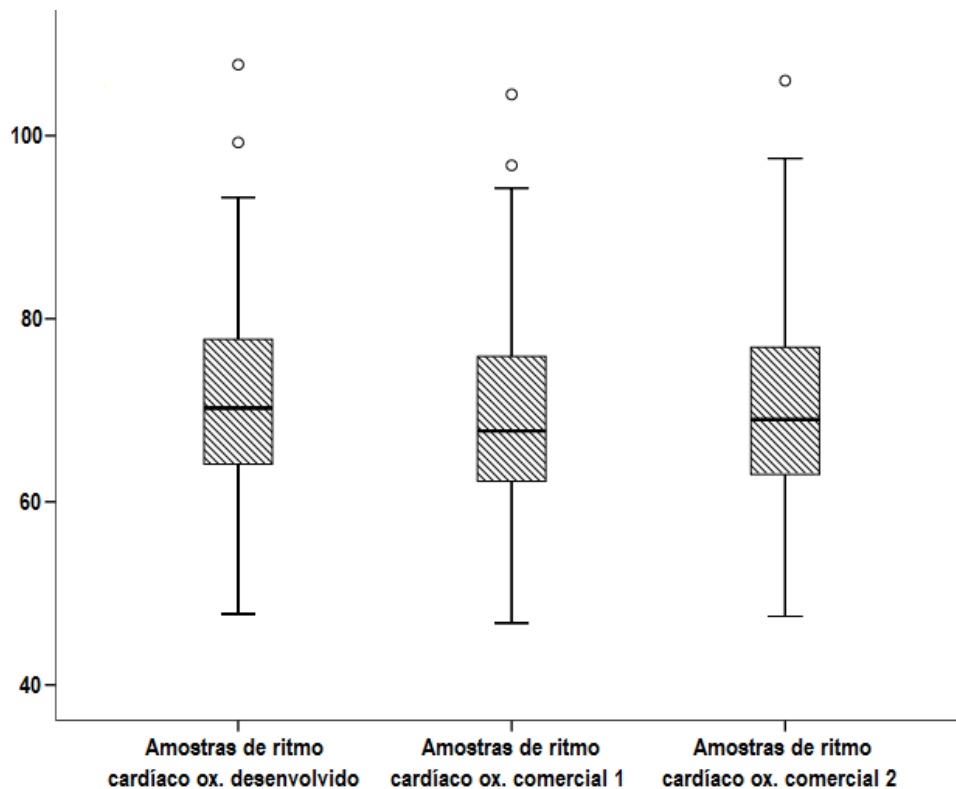


Figura 7.23: Gama das amostras de ritmo cardíaco dos três sistemas.

A tabela 7.4 fornece a informação estatística de cada oxímetro utilizado relativamente às amostras de ritmo cardíaco recolhidas, como a amostra mais elevada, mais baixa, e os percentis 25, 50 e 75.

Tabela 7.4: Estatísticas sobre as amostras de ritmo cardíaco.

	HR_Dv - amostras ritmo cardíaco ox. desenvolvido	HR_C1 - amostras ritmo cardíaco ox. comercial 1	HR_C2 - amostras ritmo cardíaco ox. comercial 2
Nº amostras	35	35	35
Mínimo	47.75	46.75	47.50
Máximo	107.75	104.50	106.00
Percentil 25	63.25	61.75	62.75
Percentil 50	70.25	67.75	69.00
Percentil 75	78.00	76.50	77.50

Através das representações gráficas da figura 7.24 é possível observar a forte força de correlação positiva entre as amostras de ritmo cardíaco dos três sistemas.

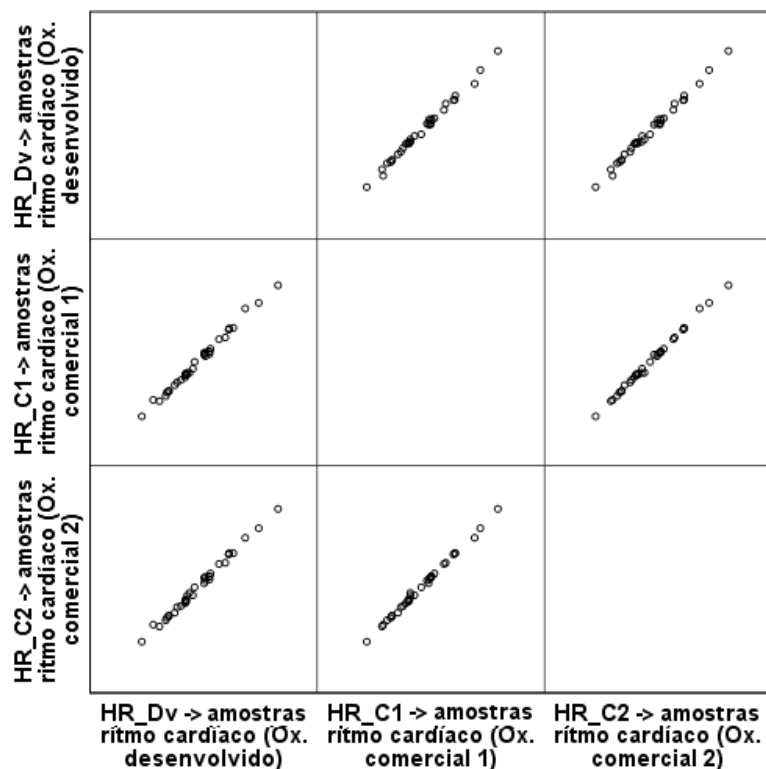


Figura 7.24: Gráfico da dispersão dos resultados de ritmo cardíaco dos três sistemas.

Os valores de correlação de Pearson entre as amostras de ritmo cardíaco dos três sistemas podem ser confirmados a partir da tabela 7.5

Tabela 7.5: Correlações de ritmo cardíaco entre os três sistemas.

Nº amostras 35	HR_Dv - amostras ritmo cardíaco ox. desenvolvido	HR_C1 - amostras ritmo cardíaco ox. comercial 1	HR_C2 - amostras ritmo cardíaco ox. comercial 2
HR_Dv amostras ritmo cardíaco ox. desenvolvido correlação de Pearson	1	0.997	0.997
HR_C1 amostras ritmo cardíaco ox. comercial 1 correlação de Pearson	0.997	1	0.999
HR_C2 amostras ritmo cardíaco ox. comercial 2 correlação de Pearson	0.997	0.999	1

A figura 7.25 apresenta a gama de valores das amostras de S_pO_2 recolhidas ao grupo de amostragem com o sistema desenvolvido, oxímetro de teste 1 e oxímetro de teste 2.

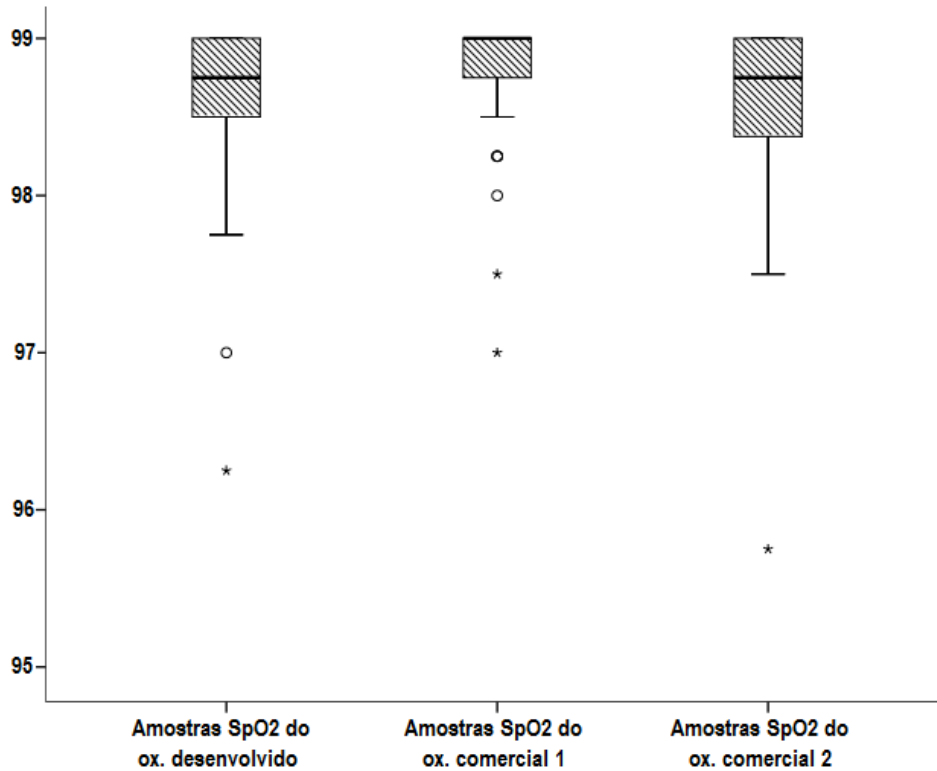


Figura 7.25: Gama das amostras de S_pO_2 dos três sistemas.

A tabela 7.6 fornece informação estatística sobre cada oxímetro utilizado relativamente às amostras de S_pO_2 recolhidas, como a amostra mais elevada, mais baixa, e os percentis 25, 50 e 75.

Tabela 7.6: Estatísticas sobre as amostras de S_pO_2 .

	SpO2.Dv - amostras SpO2 ox. desenvolvido	SpO2.C1 - amostras SpO2 ox. comercial 1	SpO2.C2 - amostras SpO2 ox. comercial 2
Nº amostras	35	35	35
Mínimo	96.25	97.00	95.75
Máximo	99.00	99.00	99.00
Percentil 25	98.50	98.75	98.25
Percentil 50	98.75	99.00	98.75
Percentil 75	99.00	99.00	99.00

Através das representações gráficas da figura 7.26 é possível observar a força de correlação positiva entre as amostras de S_pO_2 dos três sistemas.

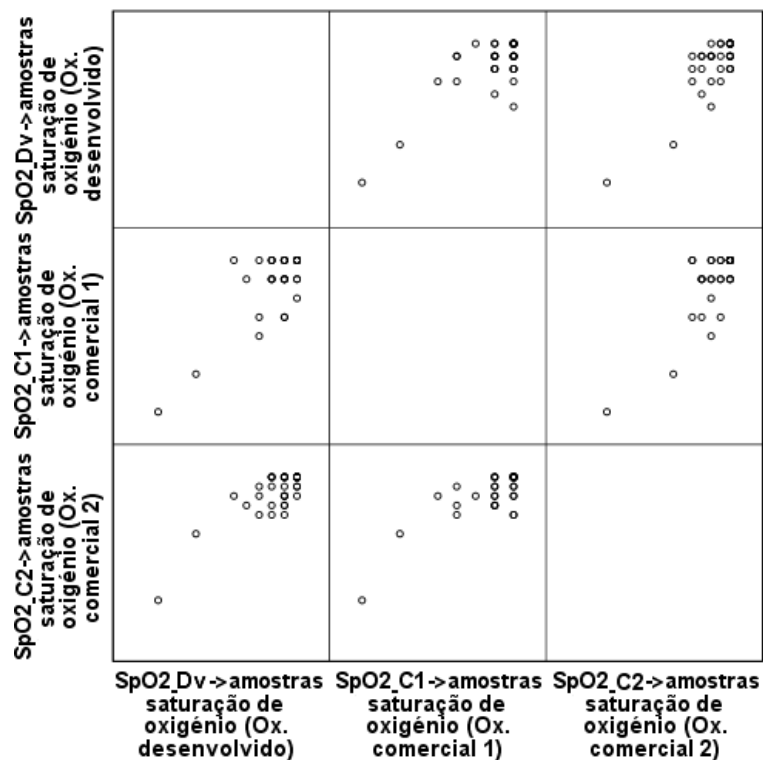


Figura 7.26: Gráfico da dispersão dos resultados de S_pO_2 dos três sistemas.

Os valores de correlação de Pearson entre as amostras de S_pO_2 dos três sistemas podem ser confirmados a partir da tabela 7.7

Tabela 7.7: Correlações de S_pO_2 entre os três sistemas.

Nº amostras 35	SpO2_Dv - amostras SpO2 ox. desenvolvido	SpO2_C1 - amostras SpO2 ox. comercial 1	SpO2_C2 - amostras SpO2 ox. comercial 2
SpO2_Dv - amostras SpO2 ox. desenvolvido correlação de Pearson	1	0.760	0.822
SpO2_C1 amostras SpO2 ox. comercial 1 correlação de Pearson	0.760	1	0.794
SpO2_C2 amostras SpO2 ox. comercial 2 correlação de Pearson	0.822	0.794	1

7.3.2 Medição na testa

Como a sonda de oximetria usada por este sistema é de reflexão, é possível colocá-la em contacto com outras zonas do corpo humano para além da ponta dos dedos. A testa, por exemplo, pode ser uma zona com um elevado interesse na monitorização do ritmo cardíaco e S_pO_2 , uma vez que pode ser aplicada a um capacete ou a uma fita para monitorizar o estado clínico de um mineiro ou de um desportista, respetivamente.

A figura 7.27 ilustra o sinal que foi possível captar usando o sistema desenvolvido e a testa como ponto de extração.

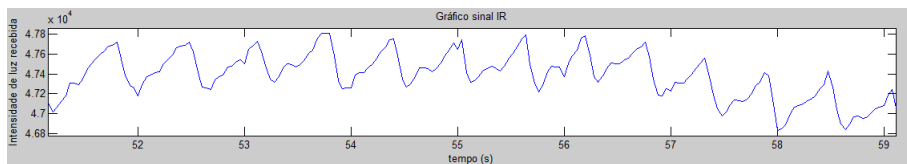


Figura 7.27: Sinal recolhido através da testa.

Analisando a figura 7.27 observa-se uma ligeira perda de qualidade no sinal recolhido em comparação com a figura 7.21, originando um possível aumento no erro relativo aos valores de ritmo cardíaco e S_pO_2 . Além disso, é necessário ajustar a sonda na testa até encontrar um local com uma boa extração de amostras, colocando em causa o princípio da oximetria de pulso, a simplicidade de utilização.

7.4 Análise de resultados

Apesar de existirem erros entre os valores finais extraídos pelo sistema desenvolvido relativamente aos dois oxímetros comerciais, os resultados são bastante positivos, uma vez que se está perante erros pequenos (menores que $\pm 2.65\%$) e pelo facto dos próprios oxímetros de teste possuírem erros associados. Além disso, o erro relativo ao S_pO_2 é extremamente baixo e praticamente igual para os dois oxímetros de teste (rondando os $\pm 0.29\%$). Por fim, deve ter-se em consideração que um oxímetro de pulso não tem a finalidade de exibir resultados 100% exatos, mas sim fornecer uma boa aproximação do estado de saúde de um paciente (normalmente usado em triagens).

Segundo a FDA [Foo15] é necessário que um oxímetro de pulso tenha no máximo um erro de $\pm 2\%$ para que possa ser considerado de excelência e colocado no mercado. O sistema do projeto teve uma resposta próxima comparativamente ao **CA-MI O2-easy**, uma vez que apenas difere nas medidas de ritmo cardíaco (em média) em $\pm 1.64\%$ e nas medidas de S_pO_2 em $\pm 0.29\%$ (dentro dos limites impostos pela FDA). Por outro lado, apesar dos resultados diferirem um pouco mais comparativamente ao **GIMA OXY-6**, continuam a ser uma boa aproximação e não se deve esquecer que tanto o **CA-MI O2-easy** como o próprio **GIMA OXY-6** tem um erro de até $\pm 2\%$ associado às suas medidas.

A fiabilidade dos resultados é comprovada pela análise estatística, nomeadamente pelas correlações entre cada sistema. Nos valores de ritmo cardíaco, os três sistemas apresentam valores extremamente elevados de correlação entre eles. O projeto desenvolvido apresenta uma correlação de 0.997 com os dois oxímetros de teste, o que significa uma extrema aproximação nas medidas relativamente a estes dois sistemas. No caso do S_pO_2 a correlação do sistema

desenvolvido é de 0.760 e 0.822 relativamente ao oxímetro comercial 1 (GIMA) e oxímetro comercial 2, respetivamente. Apesar disso são valores elevados, permitindo validar estes resultados, além de que a correlação entre os valores de S_pO_2 do sistema desenvolvido e o oxímetro comercial 2 (0.822) é superior à correlação entre as amostras do oxímetro comercial 1 e oxímetro comercial 2 (0.794).

Os resultados obtidos foram muito positivos, porém, deveria ter-se retirado valores de S_pO_2 entre 70% e 100% [Foo15] para o cálculo do erro. Tal não foi possível, uma vez que o grupo de amostragem se encontrava em excelentes condições de saúde, o que originou a extração de valores de S_pO_2 sempre maiores que 95%. Esta condição pode significar que o erro médio de S_pO_2 calculado não corresponda inteiramente à verdade. Para verificar esta situação teriam de ser realizados mais testes a indivíduos que apresentem baixos níveis de S_pO_2 ou que de algum modo se consiga reduzir artificialmente os níveis de S_pO_2 de um indivíduo.

Relativamente aos valores do ritmo cardíaco o problema acima não acontece, uma vez que o grupo de amostragem apresentou um intervalo de valores na sua pulsação entre 48 BPM e 108 BPM.

Em resumo, conseguiu-se desenvolver um sistema de oximetria, portátil, composto com módulos de baixo custo e com a precisão necessária para ser usado na constante monitorização de um indivíduo.

Capítulo 8

Conclusão e trabalho futuro

8.1 Conclusão

No planeamento deste sistema, aspetos como a portabilidade, módulos de baixo custo, protocolos de comunicação abertos e extensibilidade foram aspetos tomados em consideração. Estes objetivos foram alcançados com a elaboração de uma plataforma sem fios na qual se consegue monitorizar o ritmo cardíaco, S_pO_2 e temperatura de um indivíduo, portanto, este sistema poderá servir de base para que futuros projetos possam ser desenvolvidos na vertente de monitorização constante de sinais vitais, nos mais variados cenários.

O desempenho do sistema foi de acordo ao esperado inicialmente, uma vez que apresenta erros baixos, na ordem dos $\pm 0.28\%$ no S_pO_2 e $\pm 2.65\%$ no ritmo cardíaco, comparativamente ao **GIMA OXY-6**. Relativamente ao **CA-MI O2-easy**, o sistema apresenta um erro de $\pm 0.29\%$ no S_pO_2 e $\pm 1.64\%$ no ritmo cardíaco. Os dados extraídos do sistema acerca do ritmo cardíaco e S_pO_2 são muito satisfatórios, uma vez que apresentam níveis de correlação muito elevados relativamente aos dois oxímetros de teste, possibilitando a utilização de um sistema deste género na constante monitorização de um indivíduo de uma forma simples e fiável.

Analisando o atual estado do projeto e o trabalho desenvolvido ao longo do ano académico, a apreciação pessoal deste trabalho é bastante positiva, uma vez que os objetivos principais foram atingidos com a criação de um sistema preciso de aquisição de dados relativos ao ritmo cardíaco, S_pO_2 e temperatura, juntamente com uma aplicação *Android* para processamento e visualização de todos os dados. Além disso, adquiriu-se de forma autónoma experiência multidisciplinar relativa a sistemas embutidos (integração de sensores), desenvolvimento de *software* para vários microcontroladores e criação de uma aplicação *Android*.

8.2 Trabalho futuro

Apesar do trabalho desenvolvido ter atingido os objetivos principais deste projeto, há ainda características que poderiam ser melhoradas/implementadas.

O sistema desenvolvido poderia ser alvo de uma avaliação acerca dos seus consumos energéticos, como a otimização do tempo ativo do módulo de oximetria.

Com o objetivo de aumentar a robustez do protótipo e como atualmente o sistema está montado numa placa branca, deveria ser desenvolvida uma placa PCB (*printed circuit board*) que incorporasse todos os elementos do sistema.

O desenvolvimento de um algoritmo que de alguma forma tivesse a percepção de estados

críticos de saúde do paciente através da informação recebida seria de extrema importância, uma vez que com a ativação de alarmes em tempo real, reduziria o tempo de resposta a essas situações.

Quanto à aplicação *Android*, poderia ser implementada uma função de armazenamento dos dados adquiridos, uma vez que poderia ser de extrema importância revelar estes dados mais tarde a um profissional de saúde ou até mesmo para o utilizador consultar o seu historial clínico. Além disso, a usabilidade da aplicação *Android* deveria ser alvo de melhorias.

Bibliografia

- [And16] Android Studio. <https://developer.android.com/studio/intro/index.html>. Consultado em: 16 de Setembro, 2016.
- [Ass16] American Heart Association. Heart disease, stroke and research statistics at-a-glance. pages 1–5, 2016.
- [Blu16] Bluetooth. <https://www.bluetooth.com/what-is-bluetooth-technology>. Consultado em: 2 de Setembro, 2016.
- [Cas07] Regina Frey Casiday.R. Hemoglobin and the Heme Group : Metal Complexes in the Blood for Oxygen Transport. pages 1–13, 2007.
- [CM11] CA-MI. O2-easy, 2011.
- [Dan14] R. Dano. Health and Fitness Wearables: Affecting Healthy Behaviors, Moving Beyond Fashion. page 10, 2014.
- [Des16] Global Biomedical Device Design. <http://www.globdd.com/calibration>. Consultado em: 30 de Setembro, 2016.
- [Dev16] Wearable Devices. <http://www.chinabbb.com/newsInfo.html?id=300&nid=530>. Consultado em: 7 de Outubro, 2016.
- [Dre06] Russell Paul Dresher. Wearable Forehead Pulse Oximetry: Minimization of Motion and Pressure Artifacts. 2006.
- [Eva11] Dave Evans. The Internet of Things - How the Next Evolution of the Internet is Changing Everything. *CISCO white paper*, pages 1–11, 2011.
- [Fer09] Bruno Miguel Ferreira dos Santos. *Bios : Biological Input-Output Systems*. PhD thesis, Universidade de Aveiro, 2009.
- [Foo15] Food and Drug Administration. Pulse Oximeters - Premarket Notification Submissions [510(k)s]. pages 1–19, 2015.
- [Fox11] Stuart Ira Fox. Human Physiology. *McGraw-Hill*, 2011.
- [GIM15] GIMA. OXY-6. page 253, 2015.
- [GL07] Valerie GAY and Peter LEIJDEKKERS. A Health Monitoring System Using Smart Phones and Wearable Sensors. *International Journal of ARM*, 8(2):29 – 36, 2007.

- [GZN⁺12] Min Gao, Qian Zhang, Lionel Ni, Yunhuai Liu, and Xiaoxi Tang. CardioSentinal: A 24-hour Heart Care and Monitoring System. *Journal of Computing Science and Engineering*, 6(1):67–78, 2012.
- [Hew10] Hewlett Paquer. Bluetooth wireless technology basics. pages 3–5, 2010.
- [HMT⁺15] Diwei He, Stephen Morgan, Dimitrios Trachanis, Jan van Hese, Dimitris Drogoudis, Franco Fummi, Francesco Stefanni, Valerio Guarnieri, and Barrie Hayes-Gill. A Single-Chip CMOS Pulse Oximeter with On-Chip Lock-In Detection. *Sensors*, 15(7):17076–17088, 2015.
- [IB14] Reem Abedalmoniam Ismail and Sharief F Babiker. Oxygen Level Measurement Techniques : Pulse Oximetry. 2014.
- [Jav16] JavaTPoint. <http://www.javatpoint.com/android-life-cycle-of-activity>. Consultado em: 19 de Setembro, 2016.
- [JLR⁺03] Emil Jovanov, Amanda O’Donnell Lords, Dejan Raskovic, Paul G Cox, Reza Adhami, and Frank Andrasik. Stress monitoring using a distributed wireless intelligent sensor system. *IEEE engineering in medicine and biology magazine : the quarterly magazine of the Engineering in Medicine & Biology Society*, 22(3):49–55, 2003.
- [Joh97] John G Webster. *Design of Pulse Oximeters*. 1997.
- [Kay11] Misha Kay. mHealth: New Horizons for Health through Mobile Technologies. *World Health Organization*, 3:66–71, 2011.
- [Kho15] Tran Vu Anh Khoa. Wearable Smart Technologies: New Era of Technology. 2015.
- [Klo03] Irving M. Klotz. Hemoglobin-oxygen equilibria: Retrospective and phenomenological perspective. *Biophysical Chemistry*, 100:123–129, 2003.
- [LBB02] Alexandre Pinto Lima, Peter Beelen, and Jan Bakker. Use of a peripheral perfusion index derived from the pulse oximetry signal as a noninvasive indicator of perfusion. *Critical care medicine*, 30(6):1210–3, 2002.
- [IE16] Protocolo Bluetooth low Energy. <https://learn.adafruit.com/introduction-to-bluetooth-low-energy>. Consultado em: 15 de Setembro, 2016.
- [Lem16] Christina Lemanski. Pulse Oximetry for Perioperative Monitoring. *Journal of Perianesthesia Nursing*, 31(1):86–88, 2016.
- [Mai13] Heimo Mairböurl. Red blood cells in sports: Effects of exercise and training on oxygen supply by red blood cells. *Frontiers in Physiology*, pages 1–13, 2013.
- [Mar16] Global Pulse Oximeters Market. http://www.strategyr.com/MarketResearch/Pulse_Oximeters_Market_Trends.asp. Consultado em: 25 de Agosto, 2016.
- [Mas16] Masimo O3 Oximeter. <http://www.marketwired.com/press-release/masimo-announces-ce-marking-of-o3-regional-oximetry-nasdaq-masi-1871786.htm>. Consultado em: 15 de Outubro, 2016.

- [Max15] Maxim. Max30102 Datasheet. 2015.
- [MBC94] D R Marble, D H Burns, and P W Cheung. Diffusion-based model of pulse oximetry: in vitro and in vivo comparisons. *Applied optics*, 33(7):1279–85, 1994.
- [MMSL12] Fen Miao, Xiuli Miao, Weihua Shangguan, and Ye Li. MobiHealthcare System: Body Sensor Network Based M-Health System for Healthcare Application. pages 12–18, 2012.
- [Mon14] Sergi Sevil Montoro. Mobile application for obtain information from our geolocation. 2014.
- [mS16] Worldwide smartphone OS market Share. <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>. Consultado em: 12 de Novembro, 2016.
- [MZH15] Olivier Monnier, Eran Zigman, and Amit Hammer. Understanding Wireless Connectivity in the Industrial IoT. *Texas Instruments Books*, page 46, 2015.
- [Nat16] National Health Fundation of Australia. Heart arrhythmias. pages 2–7, 2016.
- [NB13] Sd Nanhore and Mm Bartere. Mobile Phone Sensing System for Health Monitoring. *International Journal*, 2(4):252–255, 2013.
- [Nei15] Neil Smyth. Android Studio Development Essentials. 2015.
- [Nel03] Nellcor. A Technology Overview of the Nellcor OxiMax Pulse Oximetry System. page 8, 2003.
- [NF09] Jim Noxon and Kristoffer Flores. SimpliciTi-compatible UART Driver. pages 1–15, 2009.
- [NHS10] NHS. Market review Pulse oximeters in primary and prehospital care. 2010.
- [NXP14] NXP. Medical Applications User Guide. pages 1–134, 2014.
- [oH16] Spectrum Absorption of Hemoglobin. <http://quantifiedhealth.blogspot.pt/2015/05/apple-watch-contains-unactivated-pulse.html>. Consultado em: 28 de Agosto, 2016.
- [Ols08] Christian Olsen. The Mobile Economy. *Adweek*, pages 1–56, 2008.
- [ONU15] ONU. World population, ageing. pages 1–146, 2015.
- [OPH⁺11] Yechiam Ostchega, Kathryn S Porter, Jeffery Hughes, Charles F Dillon, and Tatiana Nwankwo. Resting pulse rate reference data for children, adolescents, and adults: United States, 1999-2008. *National health statistics reports*, pages 1–16, 2011.
- [oT16] Internet of Things. <http://eluminoustechologies.com/internet-of-things>. Consultado em: 6 de Outubro, 2016.
- [Oxi16a] Nonin Xpod Oximeter. <http://www.calmedhawaii.com/product/DB8000Q>. Consultado em: 16 de Agosto, 2016.

- [Oxi16b] NoninConnect Pulse Oximeter. <http://www.prnewswire.com/news-releases/nonin-medical-introduces-apple-ios-compatible-noninconnect-elite-model-3240-bluetooth-smart-wireless-finger-pulse-oximeter-300285558.html>. Consultado em: 15 de Agosto, 2016.
- [Oxi16c] Oximetry. <https://www.elcaminohospital.org/library/oximetry>. Consultado em: 15 de Agosto, 2016.
- [Oxi16d] Pulse Oximetry. <http://www.pulseoximeter.org/cms50f.html>. Consultado em: 28 de Novembro, 2016.
- [Phi03] Philips Semiconductors. AN10216-01 I²C manual. page 51, 2003.
- [PM14] Grantham Pang and Chao Ma. A Neo-Reflective Wrist Pulse Oximeter. *IEEE Access*, 2:1562–1567, 2014.
- [PMD09] Matthew Brian Parlato, Jonathan Meyer, and Bogdan Dzyubak. Low Cost Pulse Oximeter Probe. 2009.
- [PPB⁺12] Shyamal Patel, Hyung Park, Paolo Bonato, Leighton Chan, and Mary Rodgers. A review of wearable sensors and systems with application in rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 9(1):21, 2012.
- [PWC14] PWC. Health wearables : Early Days. page 12, 2014.
- [QG12] Han Qi and Abdullah Gani. Research on mobile cloud computing: Review, trend and perspectives. pages 195–202, 2012.
- [RD09] Ana Rita and Carvalho Domingues. *Development of a Stand- Stand - Alone Pulse Oximeter*. PhD thesis, Universidade de Coimbra, 2009.
- [San05] Marian Sandmaier. Your Guide to Living Well With Heart Disease. *U.S. Department of Health and Human Services*, 2005.
- [SBKA09] Rifat Shahriyar, Faizul Bari, Gourab Kundu, and Sheikh Iqbal Ahamed. Intelligent Mobile Health Monitoring System (IMHMS). *International Journal of Control*, 2(3):13–28, 2009.
- [Sch08] Alan N Schechter. Hemoglobin research and the origins of molecular medicine. *Blood*, 112(10):3927–3938, 2008.
- [Sci16] ScienceDaily. <https://www.sciencedaily.com/releases/2009/03/090310084844.htm>. Consultado em: 7 de Outubro, 2016.
- [SHR93] D. Seidler, M. M. Hirschl, and G. Roeggla. Limitations of pulse oximetry. *The Lancet*, pages 1600–1601, 1993.
- [Shr15] Akash Shrivastava. Exploring the smartwatch as a tool for medical adherence (Master thesis). 2015.
- [SIG14] Bluetooth SIG. Bluetooth SIG 2014 Annual Report. 2014.

- [SK90] John W Severinghaus and S O Koh. Effect of anemia on pulse oximeter accuracy at low saturation. *Journal of clinical monitoring*, 6(2):85–88, 1990.
- [SM08] Norbert Stubán and Niwayama Masatsugu. Non-invasive calibration method for pulse oximeters. *Periodica Polytechnica*, 521(2):91–94, 2008.
- [SMR14] Hadi Salah, Emily MacIntosh, and Nirusan Rajakulendran. MaRS Market Insights Wearable Tech : Leveraging Canadian Innovation to Improve Health. *MaRS-Ontario Network of Enterpenieurs*, pages 1–45, 2014.
- [Tex13] Texas Instruments. SpO Pulse Ox Wrist Oximeter Reference Design. 2013.
- [TL15] Win Tin and Mithilesh Lal. Principles of pulse oximetry and its clinical application in neonatal medicine. *Seminars in Fetal and Neonatal Medicine*, 20(3):192–197, 2015.
- [TMSY14] Toshiyo Tamura, Yuka Maeda, Masaki Sekine, and Masaki Yoshida. Wearable Photoplethysmographic Sensors—Past and Present. *Electronics*, 3(2):282–302, 2014.
- [Vah09] Timo Vahtera. Midi interpreter software. 2009.
- [Vij89] Vijaylakshmi Kamat. Pulse oximetry. *CHEST Journal*, pages 713–715, 1989.
- [Wat06] Eric S. Watkins. The physiology and pathology of formula one Grand Prix motor racing. *Clinical neurosurgery*, 53:145–152, 2006.
- [Wor11] World Health Organization. Pulse Oximetry Training Manual. pages 1–23, 2011.
- [Yit89] Yitzhak Mendelson. Pulse oximetry. *CHEST Journal*, pages 713–715, 1989.

Apêndice A

I^2C - Bit banging

Na figura A.1 é apresentada uma ilustração de uma conexão típica I^2C entre um *master* e vários *slaves*. Em ambas as linhas é feito um *pull-up* para 5V através de uma resistência, permitindo à linha comutar entre o estado *high* e *low*.

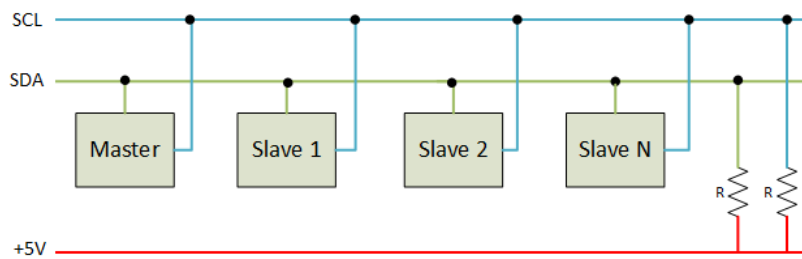


Figura A.1: Conexão I^2C comum.

A linha SDA é preenchida com bits de dados, enquanto que a linha SCL é responsável por impor o sinal de relógio. Ambas as linhas estão sempre no estado *high* (5V), a não ser que algum dispositivo faça um *pull-down* (0V). O *master* é o dispositivo que controla o sinal de relógio (SCL) e portanto é o único que pode iniciar a transferência de dados. Cada vez que um *master* pretende começar ou parar uma transferência de dados, deve enviar uma determinada sequência no barramento [Phi03]. A sequência de *start* e *stop* podem ser consultadas pela figura A.2.

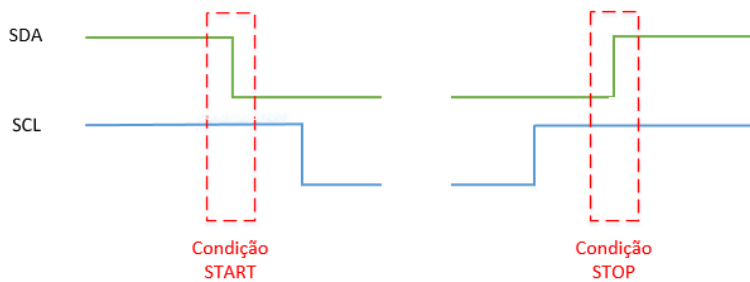


Figura A.2: Sequências de *start* e *stop*.

Depois da sequência de *start* é esperada uma transferência de dados de 8 bits começando pelo MSb (bit mais significativo), na linha SDA (figura A.3). Quando o byte for transferido, o dispositivo recetor envia de volta um bit de *acknowledgement* e dependendo se este bit transportar o estado *low* ou *high* o dispositivo transmissor terá conhecimento se o recetor está pronto ou não para receber mais dados, respetivamente [Phi03].

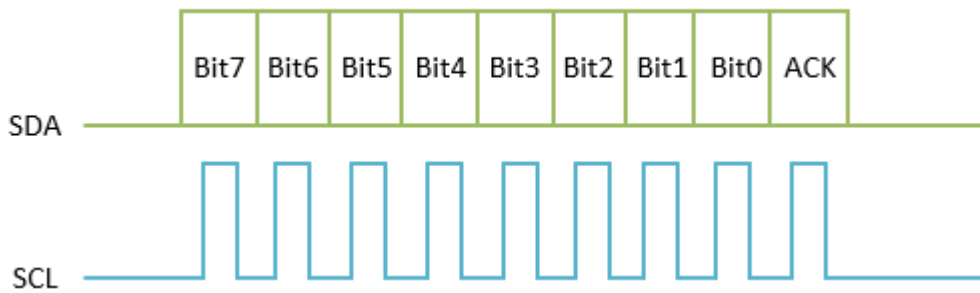


Figura A.3: SDA e SCL durante a transferência de 1 byte de dados.

O *master* pode enviar e receber dados provenientes de cada *slave*, individualmente. Isto é possível porque cada *slave* possui um endereço próprio composto por 7 bits, como pode ser visto pela figura A.4, permitindo um total de 128 conexões a *slaves* num único barramento de I^2C . Como toda a informação tem de ser enviada num byte, o oitavo bit (LSb) depois do endereço do *slave* é um bit que indica se o *master* pretende receber informação do *slave* (bit a 1) ou se o *master* deseja enviar informação para o *slave* (bit a 0) [Phi03].

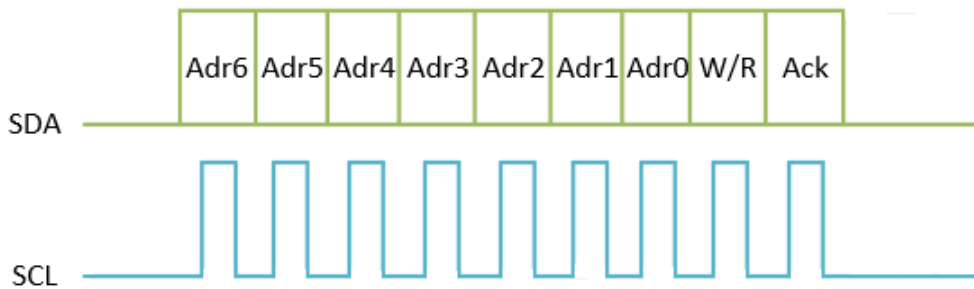


Figura A.4: SDA e SCL durante a transferência do endereço do *slave*.

Como o CC1110, utilizado na fase inicial do projeto, não possui uma interface I^2C para comunicar com os sensores, é necessário implementar este protocolo através de *software*. Este método é chamado de *bit banging* e faz uso de dois GPIOs do microcontrolador para simular um sinal de relógio e o envio de dados.

Assim, o desafio é desenvolver uma comunicação I^2C *bit banging* usando linguagem de programação C, onde o *slave* será sincronizado pela frequência de relógio do *master*. Desta forma é possível adicionar a comunicação série I^2C a qualquer microcontrolador que não possua este tipo de interface.

Apêndice B

IAR Embedded Workbench

O IAR Embedded Workbench, usado no desenvolvimento do *software* para o CC1110, é um IDE para construir e depurar aplicações em linguagem assembler, C e C++. Ao contrário do Code Composer Studio, o IAR Embedded Workbench é destinado a um enorme número de dispositivos, incluindo o CC1110. Através deste *workbench* é possível editar código, configurar o compilador, *linker* e o *debugger*.

B.1 Criação e configuração de um projeto

Para criar um novo projeto, como mostra a figura B.1, deve selecionar-se **Project** → **Create New Project**. Na janela que surgir, deve escolher-se **8051** (microcontrolador do projeto) no campo *Tool chain* e **Empty project** nos modelos de projetos. Finalmente deve clicar-se em **OK** para criar o projeto com as configurações padrão.

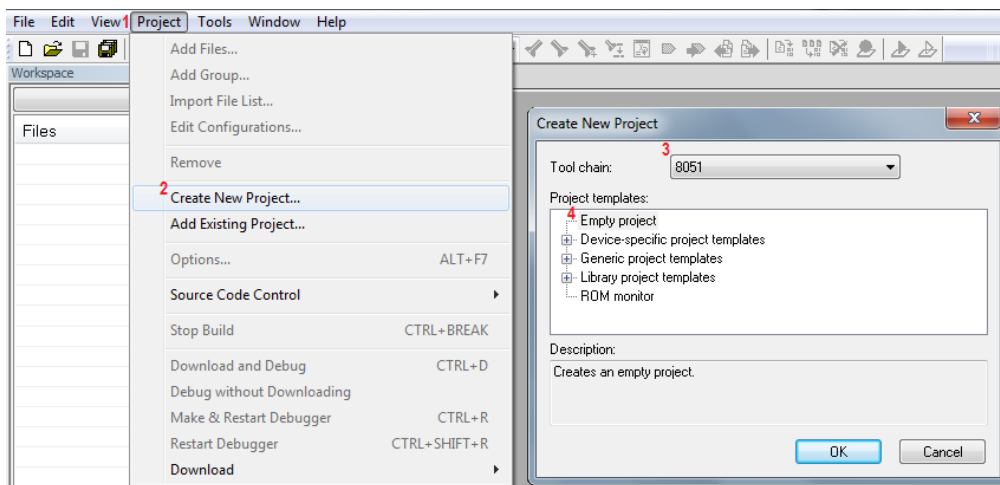


Figura B.1: IAR Workbench - *Create project*.

Porém, as opções padrão do projeto devem ser alteradas porque podem não ser adequadas para todos os microcontroladores. Para tal, deve selecionar-se **Project** → **Options** para abrir a aba General Options, como mostra a figura B.2. Sob o separador Target deve escolher-se o dispositivo com que se irá trabalhar, neste caso **CC1110F16**.

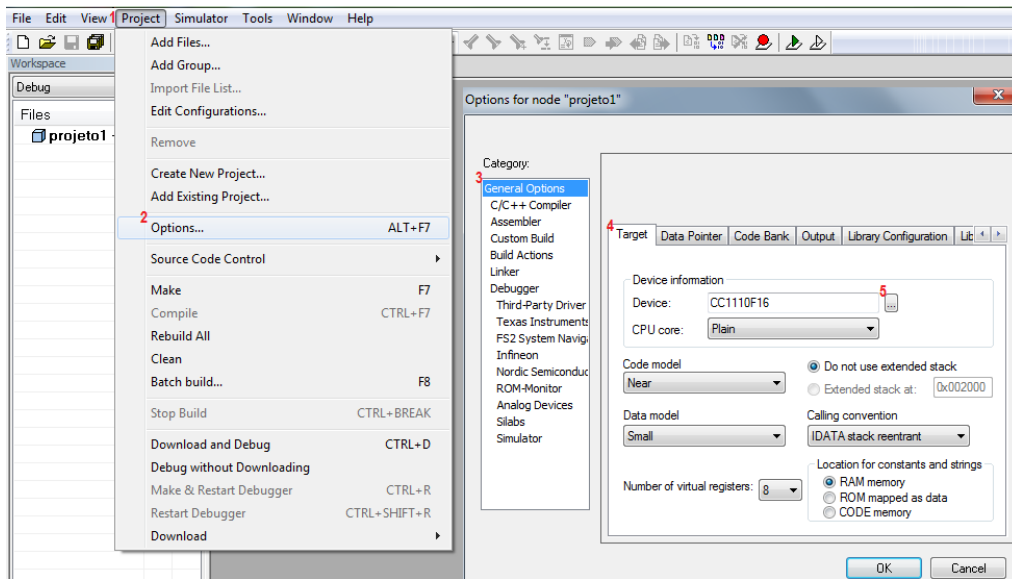


Figura B.2: IAR Workbench - *options*.

B.2 Como compilar

Depois de se desenvolver o código para o projeto é necessário compila-lo e transferi-lo para o MCU. Para tal, deve selecionar-se **Project** → **Rebuild All** para compilar e vincular o código ao CC1110. Não deve haver qualquer erro nem *warnings* na janela de mensagens durante este processo, como mostra a figura B.3.

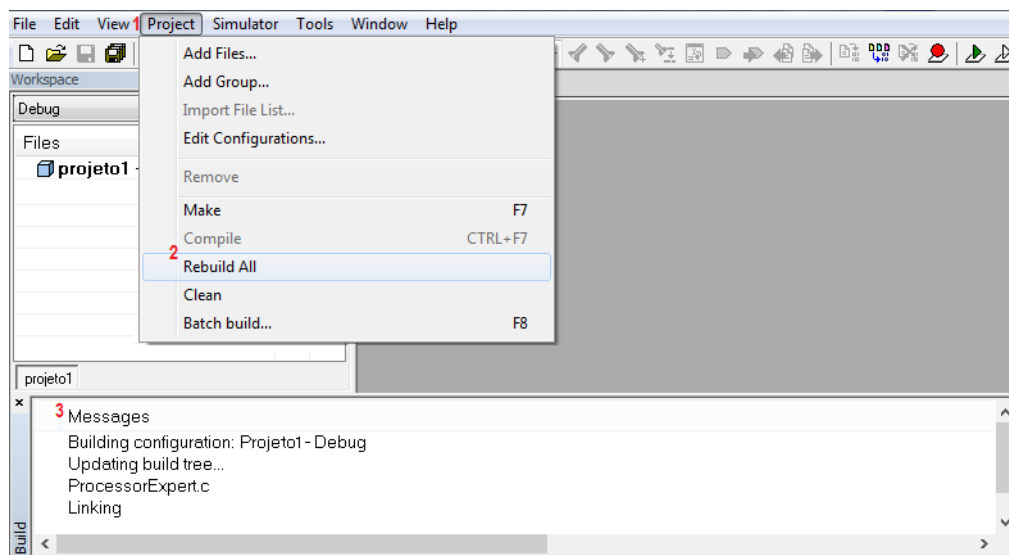


Figura B.3: IAR Workbench - *debug*.

Por fim, o programa será transferido para o CC1110 e o *debugger* estará pronto a ser usado. Para isso, deve clicar-se em **Debug** → **Go** e o *hardware* não pode ser desconectado até a operação de *debugging* ser finalizada.

Apêndice C

Sistema com MatLab

Na fase inicial decidiu-se substituir o dispositivo móvel por uma unidade de processamento dos dados no PC, o MatLab. Tirando partido da forte ferramenta MatLab, procurou-se validar inicialmente o sistema e observar de forma mais eficaz o seu desempenho.

C.1 Diagrama de blocos

O esquema de todo o sistema mantém-se praticamente o mesmo em comparação com a figura 5.1, mudando apenas os blocos onde os dados relativos aos valores ADC da hemoglobina são processados.

Temporariamente, a configuração dos blocos permaneceu de acordo com a figura C.1.

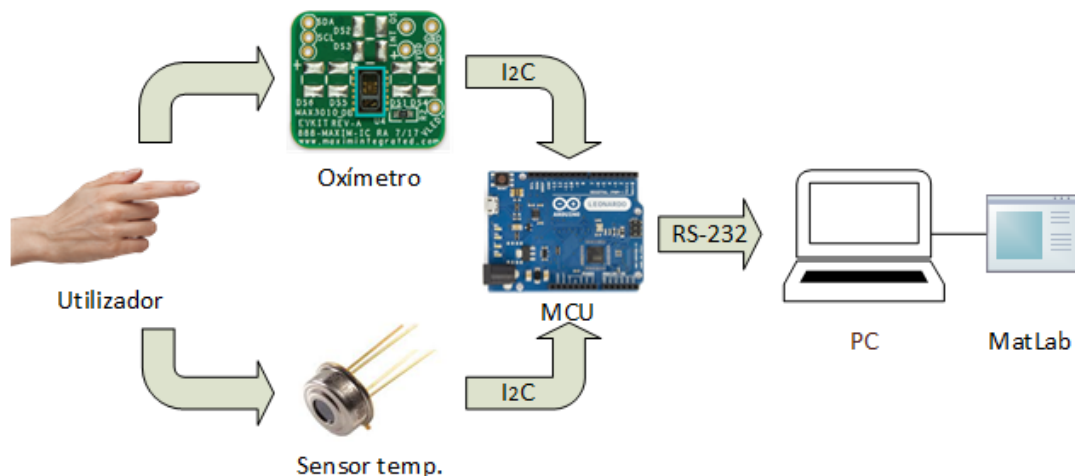


Figura C.1: Diagrama do sistema com ligação ao MatLab.

Além disso, as configurações iniciais enviadas pelo MCU aos sensores de temperatura e oximetria mantém-se, assim como as técnicas de processamento do sinal. O filtro passa-alto digital continua a ser utilizado para retirar os valores do ritmo cardíaco do utilizador e o filtro passa-baixo digital para extrair os valores de S_pO_2 .

O ficheiro **Mat_monitoring.m** contém o código MatLab desenvolvido para receber, tratar e exibir os dados referentes ao estado clínico de um utilizador.

C.2 Resultados

Depois dos dados serem processados pelo algoritmo desenvolvido, os resultados podem ser consultados em dois formatos. O primeiro fornece informações gráficas relativas à temperatura e ao atual estado da pulsação cardíaca do utilizador, uma vez que permite a análise instantânea do nível de hemoglobina presente nas artérias. A figura C.2 exibe como os resultados em forma de gráfico são apresentados ao utilizador.

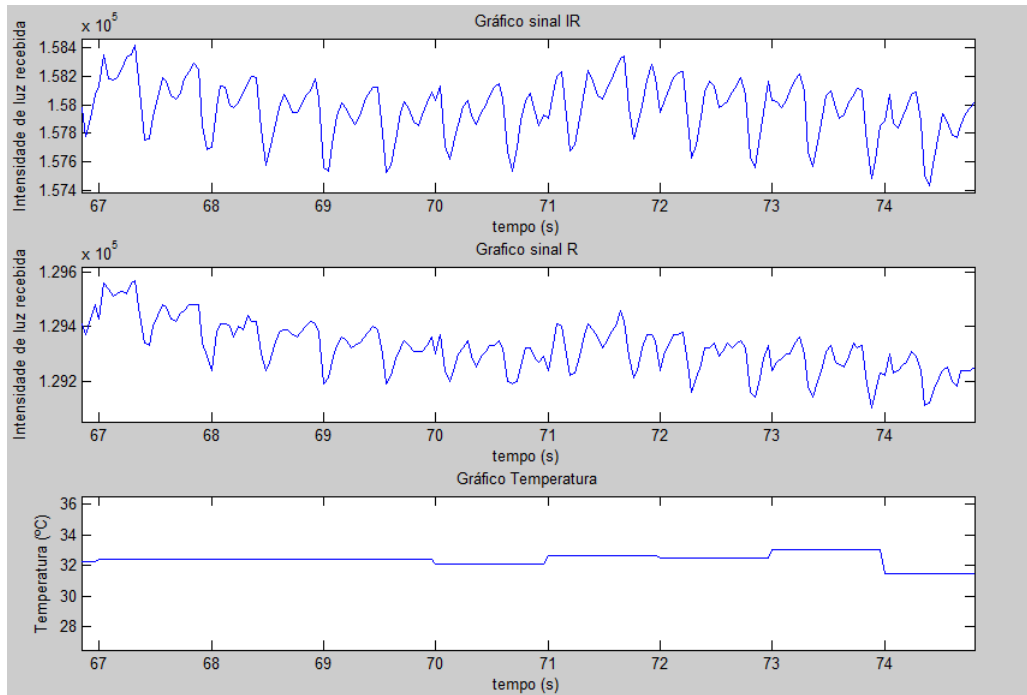


Figura C.2: Sinal cardíaco no MatLab.

Além dos gráficos é ainda apresentado na consola do MatLab o valor instantâneo da frequência cardíaca e nível de S_pO_2 , como se pode ver pela figura C.3.

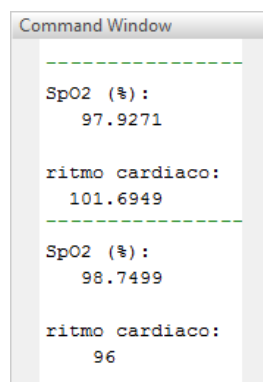


Figura C.3: Apresentação dos resultados no MatLab.

Apêndice D

Criação da aplicação *Android*

D.1 Android Studio

O Android Studio, baseado em IntelliJ IDEA, é o IDE oficial para o desenvolvimento de aplicações *Android* e por essa razão foi a ferramenta utilizada [Nei15].

A primeira tarefa a realizar é o *download* do JDK (*Java development kit*), que contém todos os componentes primários relativos ao Java, assim como ferramentas de programação. De seguida, deve-se instalar o editor (Android Studio) e por fim as ferramentas SDK, usando o SDK Manager. O SDK Manager poder ser executado através do Android Studio pelo menu **Tools** → **Android** → **SDK Manager**. Esta ferramenta, instala todos os pacotes necessários e permite ao utilizador instalar outros pacotes que ache convenientes [Nei15]. No momento do desenvolvimento da aplicação *Android* foram instalados todos os pacotes disponíveis e ainda os pacotes e ferramentas necessárias para a versão 5.0 (API 23), uma vez que o dispositivo *Android* utilizado possui essa versão.

Com a instalação do editor e das ferramentas SDK a aplicação móvel *Android* pode começar a ser desenvolvida.

D.1.1 Criação de um projeto

Na criação de um projeto deve-se começar por executar o Android Studio. Quando uma janela de boas-vindas surgir no ecrã basta clicar na opção **Start a new Android Studio project**, para exibir a janela da figura D.1.

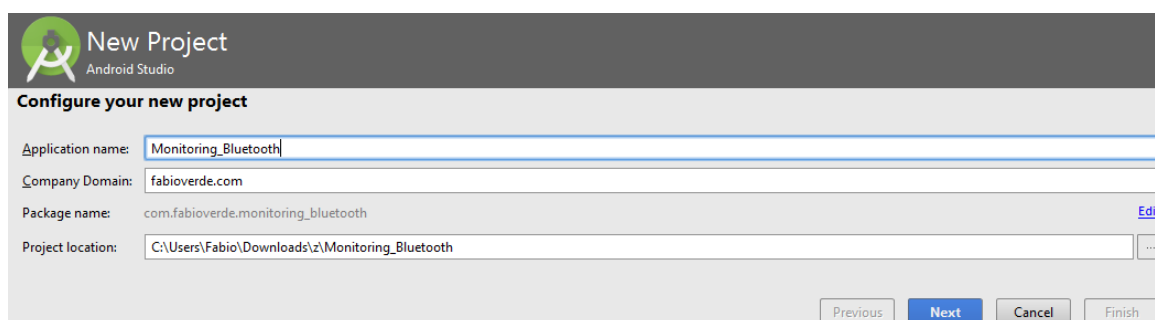


Figura D.1: Configuração de um novo projeto.

As aplicações têm um nome apresentado a todos os utilizadores, um nome de domínio e nome de pacote. Depois destes campos serem preenchidos é necessário indicar a versão *Android* a utilizar (API 23 neste projeto). Deve-se ter bastante atenção pela versão escolhida, uma vez que isso define a gama de dispositivos que podem executar a aplicação no futuro.

D.1.2 Interface com o utilizador

Depois da criação de um projeto uma janela como a da figura D.2 aparecerá, permitindo ao utilizador desenvolver qualquer aplicação.

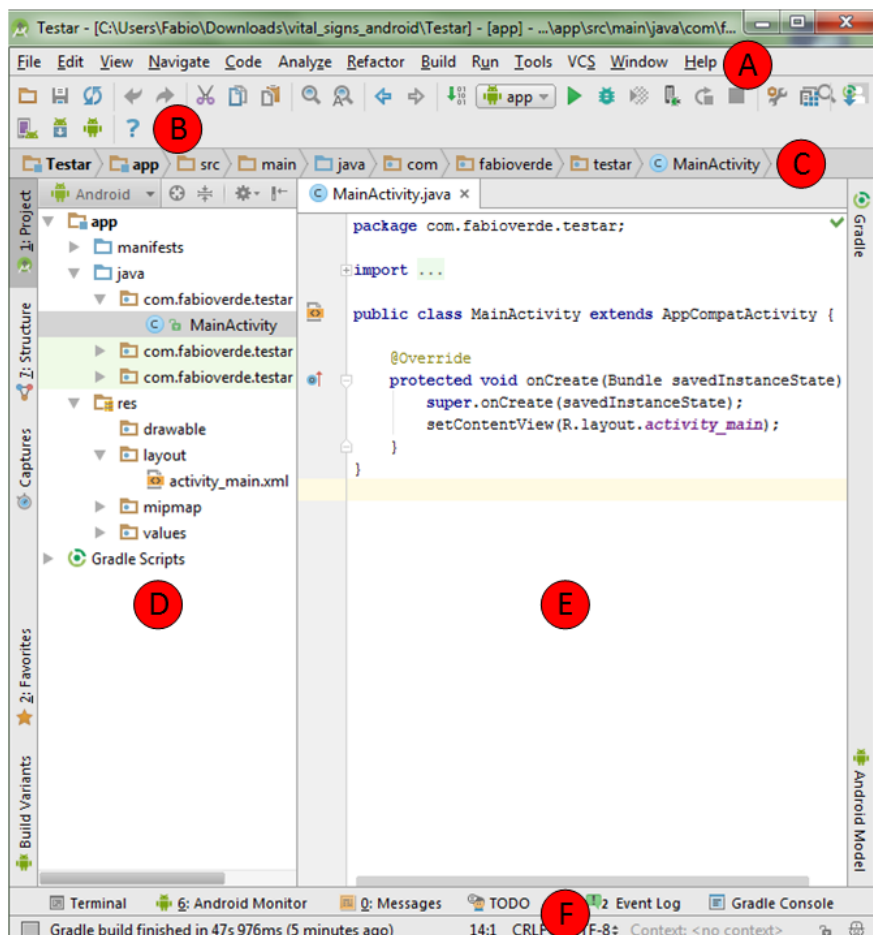


Figura D.2: *Workspace* do Android Studio.

Os elementos principais desta interface são:

- **A - Barra de Menus** - Incorpora todos os menus para executar as tarefas neste ambiente de desenvolvimento.
- **B - Toolbar** - Este elemento pode ser personalizado para fornecer um acesso mais rápido a um dado grupo de tarefas.

- **C - Barra de navegação** - Fornece ao utilizador uma forma prática de se mover entre ficheiros e pastas do projeto.
- **E - Janela do projeto** - Fornece uma vista em hierarquia de todo o projeto, permitindo o rápido acesso a um determinado ficheiro.
- **D - Janela de edição** - Exibe o conteúdo do ficheiro que o utilizador está a trabalhar (normalmente código).
- **F - Barra de Status** - Exibe mensagens com informação acerca do projeto e das atividades do Android Studio.

D.1.3 Como compilar e executar

Depois do código do projeto ser desenvolvido deve ser compilado e executado. Para tal, basta clicar no botão **run** representado por um triângulo verde situado na *toolbar* do Android Studio, como se pode ver pela figura D.3.

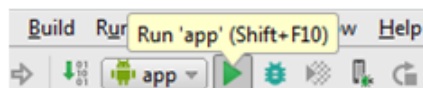


Figura D.3: Botão de *run*.

O Android Studio reagirá a esse pedido exibindo uma janela (figura D.4), onde se pode escolher o dispositivo para onde se pretende transferir a aplicação móvel. Depois de o selecionar, basta clicar em **OK** para compilar, instalar e executar a aplicação.

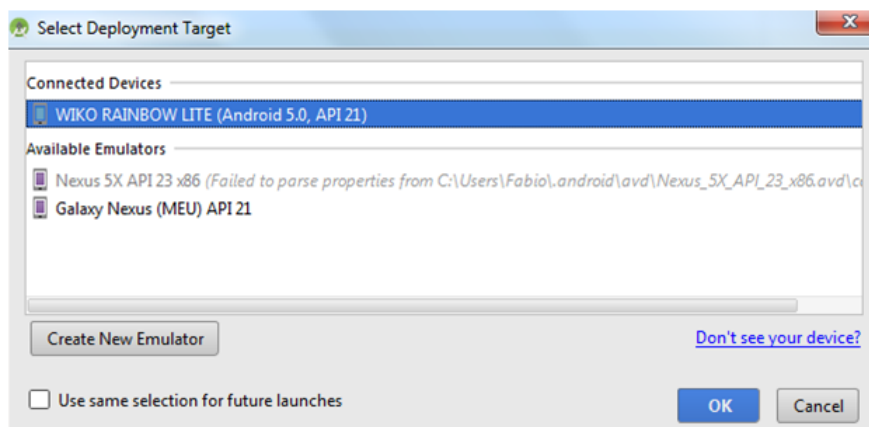


Figura D.4: Janela de escolha do dispositivo.

Neste momento, o Android studio está pronto a transferir a aplicação desenvolvida para o *smartphone*, porém, para que isto seja possível é necessário um último passo. Por defeito os dispositivos *Android* vêm configurados com um bloqueio na depuração USB, que impossibilita a instalação de aplicações provenientes de fontes como o PC. Para ativar esta função deve-se

abrir as **definições** do *smartphone* e procurar o campo **opções de programador**. Por fim deve ativar-se a opção **depuração USB** para permitir a instalação de *software* do PC.

Em alternativa a um dispositivo móvel, o Android Studio fornece a possibilidade de se executar a aplicação desenvolvida dentro do próprio ambiente de desenvolvimento, através de um emulador *Android*. Para tal, é necessário definir através do AVD (*Android virtual device*) as características do dispositivo pretendido para posteriormente ser executado pelo emulador.